

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Constantine 2- Abdelhamid Mehri



Faculté des Nouvelles Technologies de l'Information et la Communication  
Département des Technologies des Logiciels et des Systèmes d'Information

---

Projet de fin d'études pour l'obtention du diplôme de  
Licence en Informatique  
Option : Sciences de l'Informatique

Thème

**Gestion des Dons d'une Association Caritative**

Dirigé par :

Dr : Benzadri Zakaria

Réalisé par :

❖ Boufar Tarek  
❖ Chabane Youcef El  
Amine

# DEDICASES :

---

Toutes les lettres ne sauraient trouver les mots qu'il faut... Tous les mots ne sauraient exprimer la gratitude, l'amour, Le respect, la reconnaissance... Aussi, c'est tout simplement que nous dédions cette thèse a nos familles et amis pour leurs soutiens quotidiens et continue.

# REMERCIEMENTS :

---

Nous tenons avant tous à remercier monsieur le professeur Benzadri Zakaria d'avoir bien voulu diriger ce travail, de nous avoir orienté et précieusement conseillé.

Nous remercions également :

- Nous chers amis d'étude d'avoir nous aidé avec leur compétence sur cette thèse, et spécialement Ghorab Mohammed Salah Amine, Mohammed Islam Bouagga.
- Les membres du jury d'avoir accepté de juger ce travail.
- Tous les enseignants qui ont contribué à notre formation.

Merci infiniment.

## Introduction générale :

« Par la recherche du progrès et le partage des succès on dure dans l'excellence. »

\*Didier Court

Ce fichier représente la description fonctionnelle d'un système qui facilite la gestion d'une association caritative traitant les dons de vêtements. Par « **Description fonctionnelle** », on vise à citer les différents acteurs, cas d'utilisation et scénarios inclus dans le système.

La réalisation d'une tâche pareille consiste premièrement à bien analyser le sujet d'atelier (cas alternatifs, exceptions, relation entre acteurs ...) et deuxièmement de profiter d'une expérience déjà réalisée avec une association caritative de Ain Mlila nommée « El Amel » pour les soins des malades et handicapés.

Ce travail est composé de 3 chapitres à présent :

**Chapitre1** : Etude préliminaire et spécification des besoins.

**Chapitre 2** : Analyse et conception.

**Chapitre 3** : Implémentation.

# Sommaire:

## *CHAPITRE 1 : ETUDE PRELIMINAIRE ET SPECIFICATION DES BESOINS.*

---

<b>1.</b>	<b>CAHIER DE CHARGE :</b>	<b>1</b>
<b>1.1</b>	<b>Présentation de l'ensemble du projet :</b>	<b>1</b>
1.1.1	Contexte et définition du problème :	1
1.1.2	Présentation de l'ensemble de l'association :	1
1.1.3	Objectif du système :	1
<b>1.2</b>	<b>Acteur et responsabilité :</b>	<b>2</b>
1.2.1	Donateur :	2
	Permanant :	2
	Non Permanant :	2
1.2.2	Nécessiteux :	2
1.2.3	Gestionnaire du dépôt :	2
1.2.4	Administrateur :	3
<b>1.3</b>	<b>Diagramme de Contexte :</b>	<b>3</b>
<b>1.4</b>	<b>Diagramme de Packages :</b>	<b>3</b>
<b>1.5</b>	<b>Maquette :</b>	<b>4</b>
	Sautentifier :	5
	Créer Compte :	5
	Donateur Compte :	6
	Accueil :	7
<b>1.6</b>	<b>Méthodologie :</b>	<b>8</b>
<b>2.</b>	<b>ELABORATION DU DIAGRAMME DE CAS D'UTILISATION :</b>	<b>8</b>
<b>2.1</b>	<b>Contrainte Système :</b>	<b>8</b>
<b>2.2</b>	<b>Diagramme d'utilisation :</b>	<b>10</b>
<b>3.</b>	<b>DESCRIPTION TEXTUELLE ET DIAGRAMME DE SEQUENCE SYSTEME : ..</b>	<b>11</b>
<b>3.1</b>	<b>Le Cas d'utilisation « S'authentifier » :</b>	<b>12</b>
<b>3.2</b>	<b>Le Cas d'utilisation « Créer Compte » :</b>	<b>13</b>
<b>3.3</b>	<b>Le Cas d'utilisation « Promettre une donation » :</b>	<b>15</b>
<b>3.4</b>	<b>Le cas d'utilisation « Ajouter un don » :</b>	<b>17</b>
<b>3.5</b>	<b>Le Cas d'utilisation « Etablir une demande » :</b>	<b>18</b>
<b>3.6</b>	<b>Le Cas d'utilisation « Valider Entrée » :</b>	<b>20</b>
<b>3.7</b>	<b>Le Cas d'utilisation « Valider Sortie » :</b>	<b>22</b>
<b>3.8</b>	<b>Le Cas d'utilisation « Effectuer un Transfert » :</b>	<b>24</b>
<b>3.9</b>	<b>Le Cas d'utilisation « Valider Transfert » :</b>	<b>25</b>

3.10 Le Cas d'utilisation « Valider Compte nécessaireux » :	26
3.11 Le Cas d'utilisation « Annuler inscription » :	28
4. CONCLUSION :	29

## **CHAPITRE 2 : ANALYSE ET CONCEPTION**

---

1. INTRODUCTION .....	30
2. DIAGRAMME DE CLASSE : .....	30
3. MODELE DE DIAGRAMME D'ACTIVITE : .....	31
3.1 Donateur : .....	31
3.2 Nécessiteux : .....	32
4. DESCRIPTION DE LA BASE DE DONNEES : .....	33
4.1 Règles de passage : .....	33
4.1.1 Classe Association : .....	33
4.1.2 Composition : .....	34
4.1.3 Généralisation : .....	34
4.2 Schéma de la base de données : .....	35
5. CONCLUSION : .....	35

## **CHAPITRE 3 : IMPLEMENTATION**

---

1. INTRODUCTION : .....	36
2. ENVIRONNEMENT DE TRAVAIL : .....	36
<input type="checkbox"/> Plateforme matérielle : .....	36
<input type="checkbox"/> Plateforme Logicielles : .....	36
3. PRESENTATION DES LANGAGES, TECHNOLOGIES ET FRAMEWORK UTILISEES : .....	37
4. REGLES DE PASSAGE : .....	39
<input type="checkbox"/> Choix du modèle MVC : .....	39
<input type="checkbox"/> Choix des servlets : .....	39
<input type="checkbox"/> Implémentation en SQL : .....	39
5. PRESENTATION DE L'APPLICATION : .....	40
<input type="checkbox"/> Modèle de l'application : .....	40

□	Le Responsive Web design : .....	40
□	Capture écrans :.....	41
5..1	Authentification : .....	41
5..2	Profil Admin :.....	41
□	Exemple de scénarios :.....	42
6.	CONCLUSION :.....	44
	CONCLUSION GENERALE :.....	44
	BIBLIOGRAPHIE :.....	44
□	Site : .....	44
□	Livre :.....	45

## Table des Figures :

Figure 1 Diagramme de Contexte.....	3
Figure 2 Diagramme de Packages .....	4
Figure 3 Maquette « S’authentifier ».....	5
Figure 4 Maquette « Créer Compte ».....	5
Figure 5 Maquette « Compte Donateur ».....	6
Figure 6 Maquette « Accueil ».....	7
Figure 7 Diagramme de Cas D’utilisation d’une Association Caritative.....	10
Figure 8 Diagramme Séquence Système "S'authentifier" .....	12
Figure 9 Diagramme Séquence Système "Créer Compte" .....	14
Figure 10 Diagramme de séquence système "Promettre une donation" .....	16
Figure 11 Diagramme de Séquence Système "Ajouter un Don" .....	17
Figure 12 Diagramme de Séquence Système "Etablir une demande" .....	19
Figure 13 Diagramme de Séquence Système "Valider Entrée" .....	21
Figure 14 Diagramme de Séquence Système "Valider Sortie".....	23
Figure 15 Diagramme de Séquence Système "Effectuer un Transfert" .....	24
Figure 16 Diagramme Séquence Système "Valider Transfert" .....	25
Figure 17 Diagramme Séquence Système "Valider Compte nécessaireux" .....	27
Figure 18 Diagramme Séquence Système "Annuler inscription" .....	28
Figure 19 Modèle de Domaine.....	30
Figure 20 Diagramme d'activité Scénario "Donateur" .....	31
Figure 21 Diagramme d'activité Scénario "Nécessiteux".....	32
Figure 22 Passage "classe association" .....	33
Figure 23 Passage "Composition" .....	34
Figure 24 Passage "Généralisation" .....	34
Figure 25 Schéma relationnel de la base de données .....	35
Figure 26 Capture écran exemple responsive .....	40
Figure 27 Capture écran "Authentification" .....	41
Figure 28 Capture écran "Profil Admin" .....	41
Figure 29 Scénario "Créer Dépôt" .....	42
Figure 30 Scénario "Ajouter donation" .....	43

## Table des Tableaux :

Tableau 1 Structure d'un Dépôt .....	9
Tableau 2 Fiche Descriptive "S'authentifier" .....	12
Tableau 3 Fiche Descriptive "Créer Compte" .....	13
Tableau 4 Fiche descriptive "Promettre une donation" .....	15
Tableau 5 Fiche descriptive " Ajouter don".....	17
Tableau 6 Fiche descriptive "Etablir une demande" .....	18
Tableau 7 Fiche Descriptive "Valider Entrée".....	20
Tableau 8 Fiche Descriptive "Valider Sortie".....	22
Tableau 9 Fiche Descriptive "Effectuer un Transfert" .....	24
Tableau 10 Fiche Descriptive "Valider Transfert" .....	25
Tableau 11 Fiche descriptive "Valider compte nécessaireux".....	26
Tableau 12 Fiche Descriptive "Annuler inscription" .....	28

<b>Tableau 13 Ordinateur Utilisé.....</b>	<b>36</b>
---	-----------



# *Chapitre 1 : Etude préliminaire et spécification des besoins.*

---

## **1. Cahier de charge :**

Un cahier des charges a pour fonction de formaliser un besoin afin que ce dernier soit compris par l'ensemble des acteurs impliqués dans le projet (dans l'association dans le cas échéant). Il précise les spécifications attendues.

### **1.1 Présentation de l'ensemble du projet :**

#### **1.1.1 Contexte et définition du problème :**

On a de plus en plus d'appels techniques à gérer. Dans le cas où une entreprise (ou association) doit gérer sa gestion de stock de façon que les entrées soient partagées pour les sorties, c'est par très aisé de retrouver une solution fiable pour ce travail. Surtout quand le produit est stocké dans plusieurs dépôts.

#### **1.1.2 Présentation de l'ensemble de l'association :**

L'association présenté se nomme « **AKSSINI** », sa tâche principale consiste à collecter de différents vêtements fournis par des donateurs (inscrits/ou pas) dans le système et essayer de livrer ces derniers à ses nécessiteux qui sont déjà acceptés. En se basant sur ce principe, l'association doit satisfaire tous les nécessiteux d'une façon égale pour avancer plus en avant et propager la miséricorde entre les gens.

#### **1.1.3 Objectif du système :**

Le système présent gère les dons (vêtements) entrant et sortant, on peut donc déterminer ses deux objectifs essentiels :

- Accepter les donations** fournis par les donateurs en leur offrant un compte qui leur permet de gérer leur dons (type de vêtement(s), quantité, période et heure de livraison ...) et les stockés dans des dépôts spéciaux.

- Accepter les demandes** des nécessiteux basée sur les dons stockés dans les dépôts.

Et vice versa beaucoup d'objectif doivent être traité car le bon déroulement d'une livraison de A à Z implique d'autres contraintes comme :

**-La gestion des dépôts** qui nous oblige à identifier un nouvel acteur « Gestionnaire du dépôt », qui sera responsable des entrées & sortie.

**-L'acceptation des demandes** établie par les nécessiteux et la gestion des comptes de l'association nous oblige aussi à identifier un nouvel acteur : Administrateur.

On ne va pas citer tous les cas d'utilisation essentiels du système, à présent on va se concentrer sur les acteurs et leur rôle envers l'association.

## 1.2 Acteur et responsabilité :

D'après les objectifs du système on a pu définir les acteurs principales responsables pour le bon fonctionnement du système. Un acteur est défini quand un nombre de fonctionnalité est relative entre lui-même.

### 1.2.1 Donateur :

Tant que l'objectif de l'association et d'aider les gens et effectuer des donations, aucune obligation d'être inscrit dans le système et mise au point. Cependant chaque donation est fournie soit par un donateur **Permanent** ou **Non Permanent**.

**Permanent** : A la création d'un compte, le donateur permanent crée un programme de donation qui définit le type de vêtement sa quantité et le moment de la livraison dans une période cyclique, il choisit un dépôt et livre son don lorsque le moment est arrivé.

**Non Permanent** : Tant que ce donateur n'est pas inscrit il peut donc promettre l'association un don dans une date et heure comme il veut.

### 1.2.2 Nécessiteux :

Les nécessiteux doivent d'abord faire une demande de création de compte pour pouvoir interagir dans le système. Lorsqu'un nécessiteux est accepté il pourra donc s'authentifier et demander un don en choisissant un vêtement disponible dans l'ensemble des dépôts. Chaque nécessiteux a une priorité à part, donc chaque commande a sa propre limite et sa propre priorité.

### 1.2.3 Gestionnaire du dépôt :

Il jouera le rôle de jointure entre les nécessiteux et les donateurs. En consultant le stock de son dépôt il doit valider les entrées et sorties, refuser en cas d'absence ... etc.

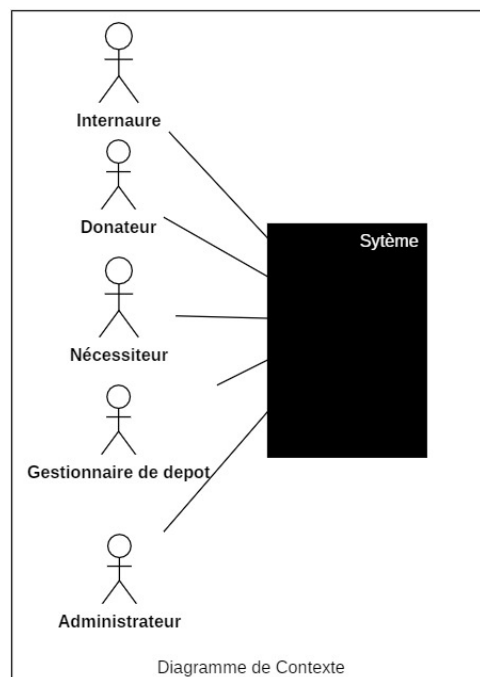
#### 1.2.4 Administrateur :

Il valide les comptes des nécessiteux après qu'ils ont effectué leurs demandes. Il peut même gérer les autres comptes, consulter le stock globale, statistique de l'association ...etc.

Une description détaillée sera établie lors des diagrammes de cas d'utilisation pour chaque acteur.

### 1.3 Diagramme de Contexte :

On commencera d'ailleurs par considérer que le futur logiciel correspond à une boîte noire qui doit fournir des services à son environnement. Par environnement, on entend les utilisateurs qui ont besoin de ce logiciel. Les acteurs sont placés à côté de la boîte noire.



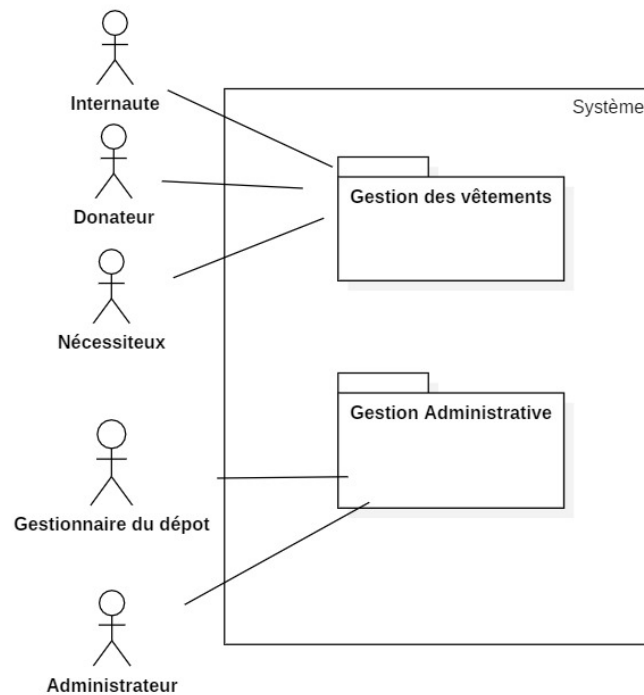
*Figure 1 Diagramme de Contexte*

### 1.4 Diagramme de Packages :

Un proverbe chinois connu dit qu'un éléphant ne peut pas être avalé d'un coup. La bête doit être coupée en morceaux si on veut en venir à bout, puis les morceaux sont cuisinés séparément avant d'être mangés. Revenons à notre projet, le but du diagramme de packages est de décomposer le logiciel à développer en plusieurs parties bien distinctes parmi toutes ses fonctionnalités. D'après notre étude on voit ces deux parties :

- La partie « **Gestion des vêtements** » qui contiendra les dons fournis et à livrer (entrées / sorties).

La partie « **Gestion administratives** » les tâches liées aux dépôts (gestion dépôts, ajouter type de vêtement, valider compte ...).



*Figure 2 Diagramme de Packages*

Nous ne voyons pas encore les liens entre les packages. Pour cela, on devra avancer un peu plus dans l'analyse.

## 1.5 Maquette :

Dans cette étape on va montrer quelques interfaces importantes (vues) de notre association.

### S'authentifier :

Représentation de l'interface responsable de l'authentification des différents utilisateurs du système.

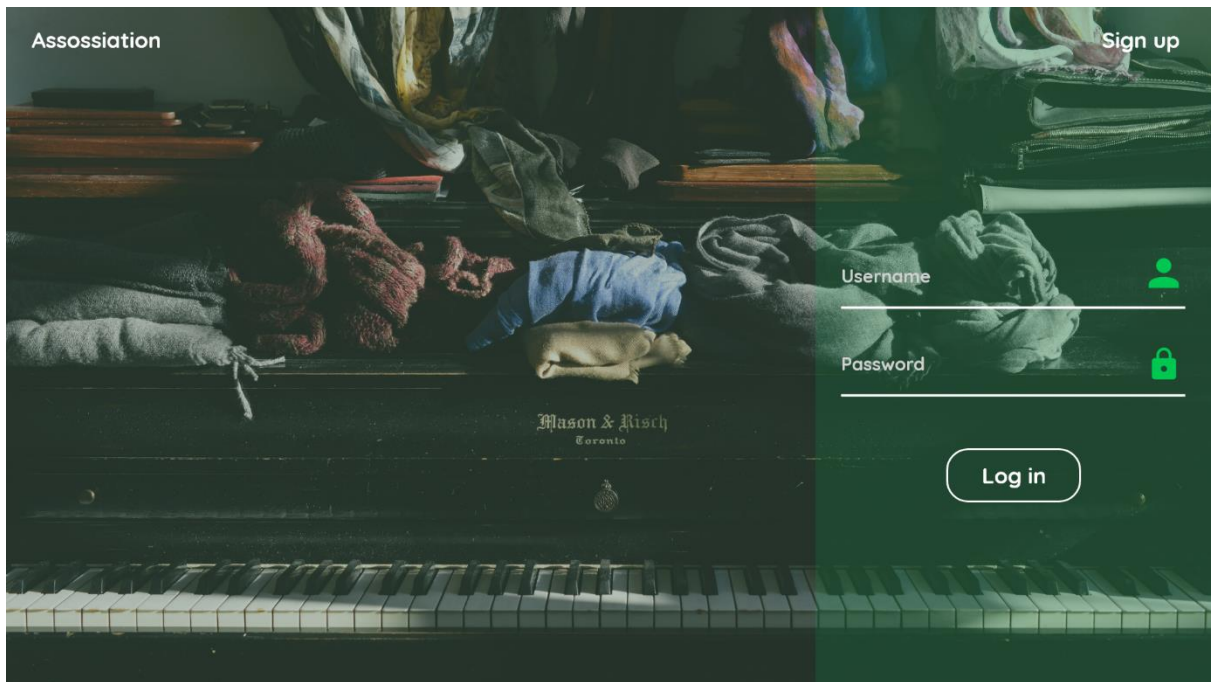


Figure 3 Maquette « S'authentifier »

### Créer Compte :

Représentation de l'interface responsable de l'inscription du donateur (inscription + programme donation).

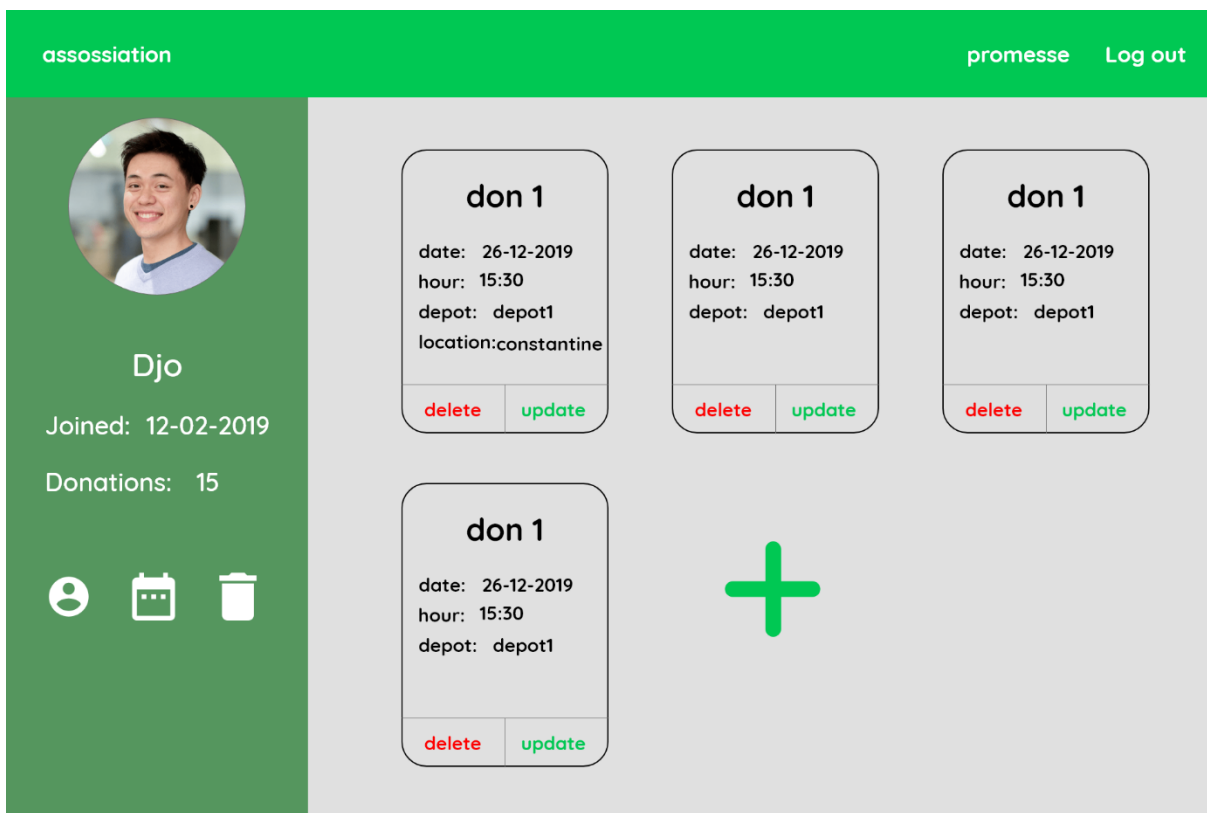


Figure 4 Maquette « Créer Compte »

## Donateur Compte :

Interface qui présente le profil du donateur.

La maquette présente une interface web pour un compte donateur. Elle est divisée en deux sections principales : 'Données Personnel' et 'Programme de Donation'. La section 'Données Personnel' contient des champs pour le nom, le prénom, la date de naissance, le sexe, le mot de passe, l'email, le téléphone et l'emploi. La section 'Programme de Donation' contient des champs pour le type de donation, la quantité, le jour et l'heure. Un bouton 'Registre' est visible en bas à droite de la section 'Programme de Donation'. Le tout est sur un fond d'image floue d'un enfant jouant avec des ballons.

Données Personnel	
Nom	Prenom
Date de Naissance	Sex
Mot de Pass	
Email	Telephone
Emplois	

Programme de Donation	
Type de donation	Quantite
Chaque	
Jour	
Heure	

Registre

*Figure 5 Maquette « Compte Donateur »*

## Accueil :

L'interface d'accueil du site.

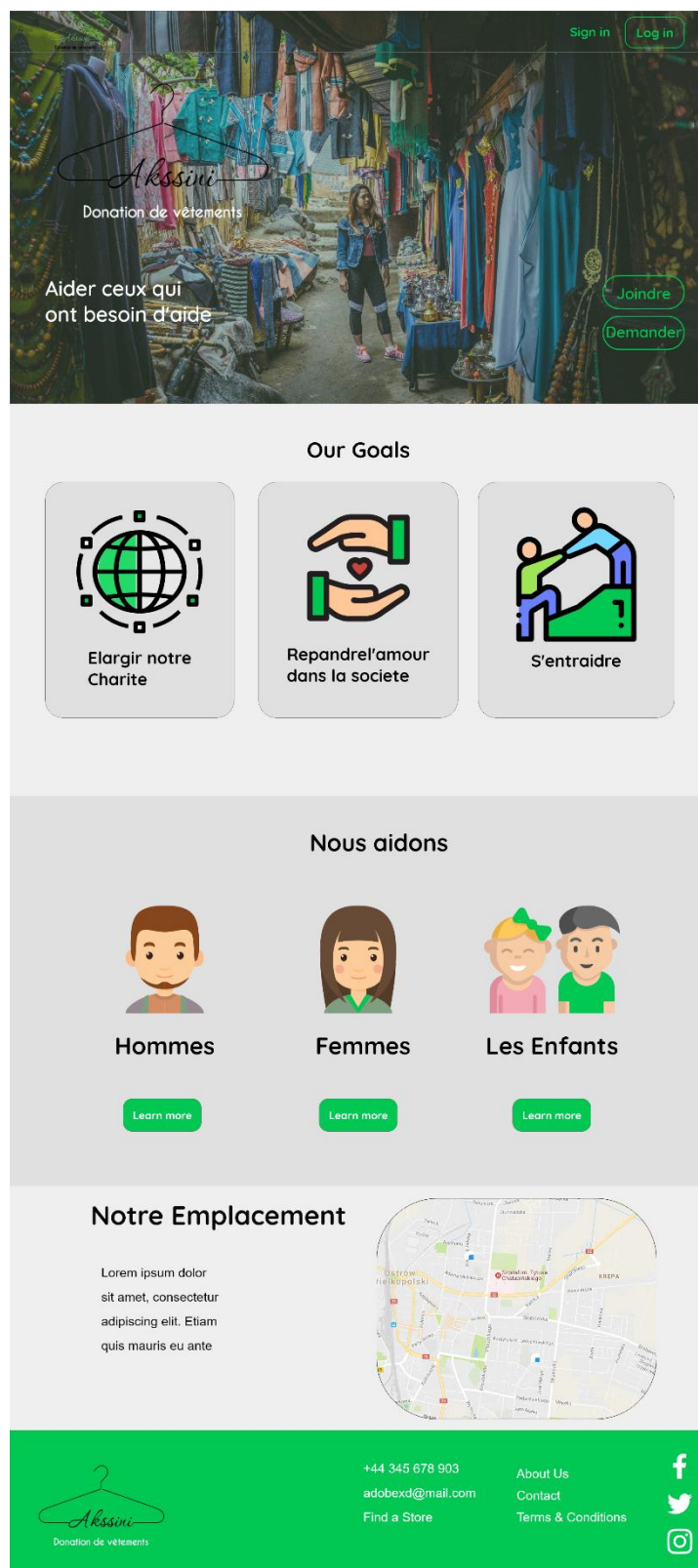


Figure 6 Maquette « Accueil »



## 1.6 Méthodologie :

Comme n'importe quel type de projet, un projet informatique nécessite une phase d'analyse, suivi d'une étape de conception. Dans la phase d'analyse, on cherche d'abord à bien comprendre et à décrire de façon précise les besoins des utilisateurs ou des clients. Que souhaitent-ils faire avec le logiciel ? Quelles fonctionnalités veulent-ils ? Pour quel usage ? Comment l'action devrait-elle fonctionner ? C'est ce qu'on appelle « **l'analyse des besoins** ».

Après validation de notre compréhension du besoin, nous imaginons la solution. C'est la partie analyse de la solution. Dans la phase de conception, on apporte plus de détails à la solution et on cherche à clarifier des aspects techniques, tels que l'installation des différentes parties logicielles à installer sur du matériel. Pour réaliser ces deux phases dans un projet informatique, nous utilisons des méthodes, des conventions et des notations. UML fait partie des notations les plus utilisées aujourd'hui.

La connaissance d'UML est donc plus que nécessaire pour continuer dans la phase d'analyse de besoins principaux des utilisateurs : **Les cas d'utilisation**.

## 2. Elaboration du diagramme de cas d'utilisation :

Pour but de faire comprendre l'idée clairement, on va citer quelques contraintes qui influent sur le Système d'information de l'association, pour but de clarifier notre système. Ces ajouts nous aident à assurer le bon fonctionnement du système (maintenance, sécurité, ordre...).

### 2.1 Contrainte Système :

-Chaque nécessaire doit d'abord faire une demande d'adhésion avant d'avoir un compte pour but de bien traiter son cas et savoir s'il désire vraiment les biens de l'association. Cette demande sera consultée par l'administrateur et c'est lui qu'il la valide ou non.

-Les donateurs qui veulent être permanents dans l'association doivent en moins effectuer une donation dans leur inscription (qui est inclus dans leur Sign In).

-Une fois une donation est effectuée dans un dépôt, un emplacement qui correspond au type du don est réservé pour le donateur (état = « Réservé »). Quand le donateur apporte son don, le gestionnaire de dépôt valide le reçu et l'état devient occupé (état = « Occupé »).



-Même chose pour la demande du nécessaire. Un nécessaire qui effectue une demande veut dire un emplacement avec un état « Occupé » est « Réserve » à lui.

-Chaque dépôt se compose de **n** chambres, chaque chambre a un type défini de don (par exemple : Chaussure, pantalon...). Chaque chambre a sa part se compose de **n** emplacement, un ensemble d'eux est attribué au permanent et d'autres aux non Permanent. Chaque ensemble de ces derniers peut être de type Homme ou Femme.

Dépôt	Dépôt 1							
Chambre	« Chaussure »				« T-shirt »			
Contrat	Permanent		Non Permanent		Permanent		Non Permanent	
Sexe	Homme	Femme	Homme	Femme	Homme	Femme	Homme	Femme
Nombre Emplacement	15	15	10	10	15	15	10	10

*Tableau 1 Structure d'un Dépôt*

**Remarque :** L'exemple ci-dessous montre une composition particulière du dépôt.

\*Ce dépôt contient 100 emplacements en tout.

## 2.2 Diagramme d'utilisation :

Représentation sous UML de **qui doit pouvoir quoi ?**



Figure 7 Diagramme de Cas D'utilisation d'une Association Caritative

### 3. Description textuelle et diagramme de séquence système :

Les diagrammes réalisés jusqu'à maintenant (diagramme de contexte, diagramme de packages, diagramme de cas d'utilisation) nous ont permis de découvrir petit à petit les fonctionnalités (appelées aussi des cas d'utilisation) que l'on devrait avoir dans le futur logiciel.

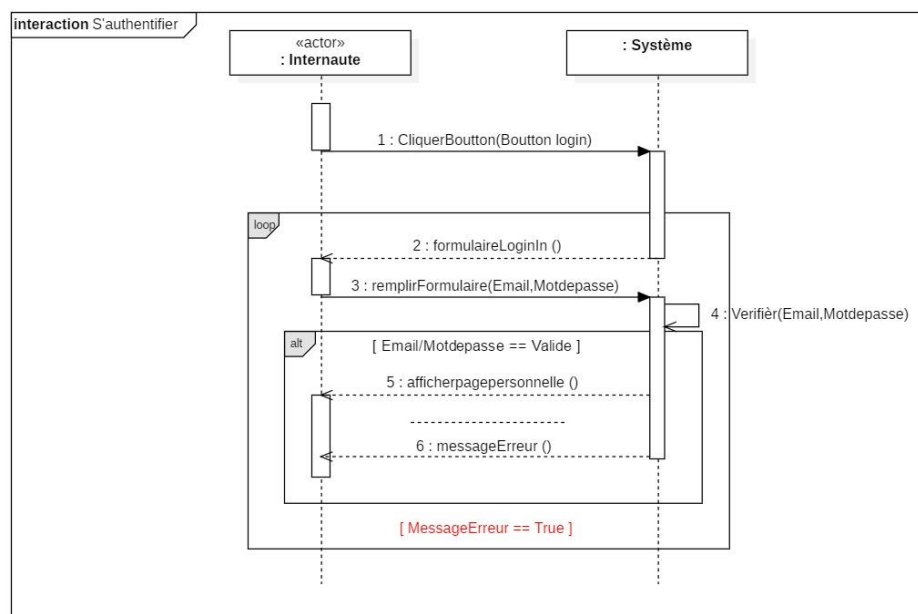
Nous allons désormais parler de l'interaction entre les acteurs et le système : il s'agit de décrire la chronologie des actions qui devront être réalisées par les acteurs et par le système lui-même. On parle d'ailleurs de scénarios.

Il s'agit maintenant d'éditer les **fiches descriptives** et les « **Diagrammes d'interactions** » de quelques cas d'utilisation essentiels du système.

### 3.1 Le Cas d'utilisation « S'authentifier » :

Propriétés	Description
<b>Nom du cas</b>	S'authentifier
<b>Acteurs</b>	Utilisateur
<b>Pré Conditions</b>	Compte déjà Créé.
<b>Post Condition</b>	Utilisateur authentifié.
<b>Scénario nominale</b>	<p>1/ L'utilisateur clique sur le bouton « Se Connecter ».</p> <p>2/ Le Système demande à l'utilisateur d'entrer son adresse Email et son mot de passe.</p> <p>3/ L'utilisateur entre ces derniers et clique sur le bouton « Se Connecter ».</p> <p>4/ Le Système renvoie l'utilisateur vers sa page personnelle.</p>
<b>Scénario alternatif</b>	<p>4.1/ Le système détecte une erreur (nom d'utilisateur ou mot de passe erroné).</p> <p>4.2/Retourner vers 2.</p>

*Tableau 2 Fiche Descriptive "S'authentifier"*



*Figure 8 Diagramme Séquence Système "S'authentifier"*

### 3.2 Le Cas d'utilisation « Créer Compte » :

Propriétés	Description
Nom du cas	Créer Compte.
Acteurs	Utilisateur.
Pré Conditions	Être En ligne (dans le site).
Post Condition	Nouveau Compte créer dans le système.
Scénario nominale	<p>1/L'utilisateur clique sur le bouton « <b>INSCRIPTION</b> ».</p> <p>2/Le Système demande à l'utilisateur s'il veut s'inscrire en tant que « Donateur » ou « Nécessiteux ».</p> <p>3/L'utilisateur choisit « Donateur ».</p> <p>4/Le système demande à l'utilisateur de remplir un formulaire.</p> <p>5/ L'utilisateur rempli le formulaire et clique sur « Suivant ».</p> <p>6/Le Système affiche un deuxième formulaire qui concerne les <u>dons</u>.</p> <p>7/L'utilisateur remplit le formulaire avec son programme de donation voulu et clique sur le bouton « Suivant ».</p> <p>8/Le Système affiche à l'utilisateur la liste des <u>dépôts</u> et lui demande de choisir un pour le programme.</p> <p>9/L'utilisateur choisi un dépôt et clique sur « S'inscrire ».</p> <p>10/Le système enregistre les données du donateur et lui crée un compte en le renvoyant vers sa page personnelle.</p>
Scénario alternatif	<p>3.1/ L'utilisateur choisi « Nécessiteux ».</p> <p>3.2/Le Système demande à l'utilisateur d'entrer ses informations personnelles (tous ce qui concerne sa situation financière ...).</p> <p>3.3/L'utilisateur entre ses informations et envoie sa demande d'adhésion.</p> <p>3.4/Le système enregistre la demande, et envoie un email à l'utilisateur. (Msg de succès)</p>
Scénario d'exception	<p>10. e.1/Tous les dépôts sont saturés.</p> <p>10. e.2/Le système affiche un msg de regéré.</p>

Tableau 3 Fiche Descriptive "Créer Compte"

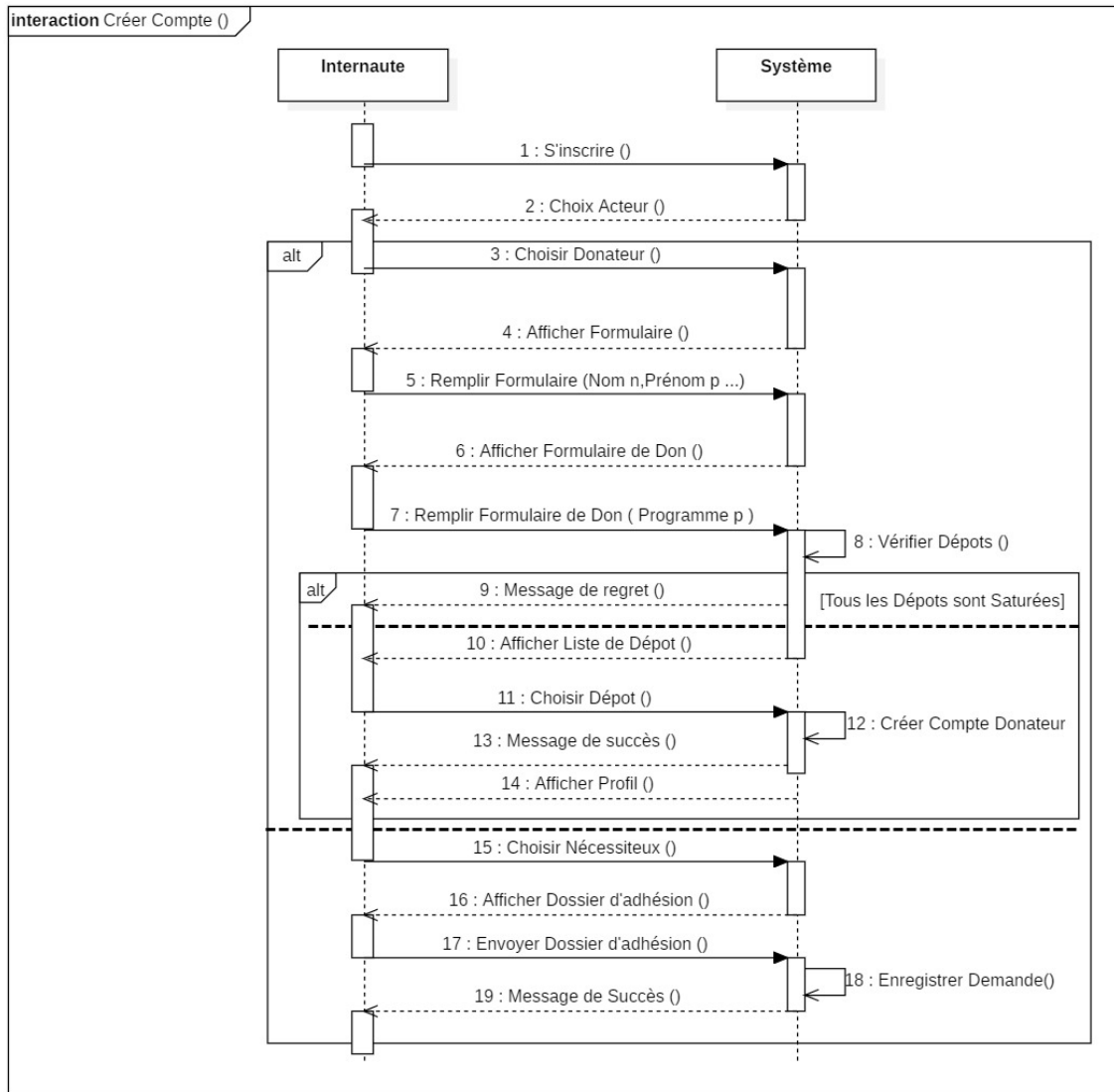


Figure 9 Diagramme Séquence Système "Créer Compte"

### 3.3 Le Cas d'utilisation « Promettre une donation » :

Propriétés	Description
<b>Nom du cas</b>	Promettre une donation.
<b>Acteurs</b>	Internaute.
<b>Pré Conditions</b>	Être En ligne (dans le site).
<b>Post Condition</b>	Enregistrement de la demande établie.
<b>Scénario nominale</b>	<p>1/L'utilisateur clique sur le bouton « Promettre une Donation ».</p> <p>2/ Le système demande à l'utilisateur d'entre ses informations personnelles.</p> <p>3/ L'utilisateur entre ces derniers.</p> <p>4/ Le Système demande à l'utilisateur d'entrer le type de don, sa quantité, la date et le temp de la livraison.</p> <p>5/L'utilisateur entre sa demande.</p> <p>6/ Le système affiche à l'utilisateur la listes des dépôts disponibles pour sa demande.</p> <p>7/ L'utilisateur sélectionne le dépôt voulu.</p> <p>8/ Le système enregistre la promesse en affichant a l'utilisateur que sa demande est réussite.</p>
<b>Scénario alternatif</b>	<p>2.1/L'utilisateur est déjà connecté à son compte. (Cas : donateur permanent, gestionnaire de stock...).</p> <p>2.2/Aller à 4.</p>
<b>Scénario d'exception</b>	<p>6. e.1/Tous les dépôts sont saturés.</p> <p>6. e.2/Le système affiche un message de regret.</p>

*Tableau 4 Fiche descriptive "Promettre une donation"*

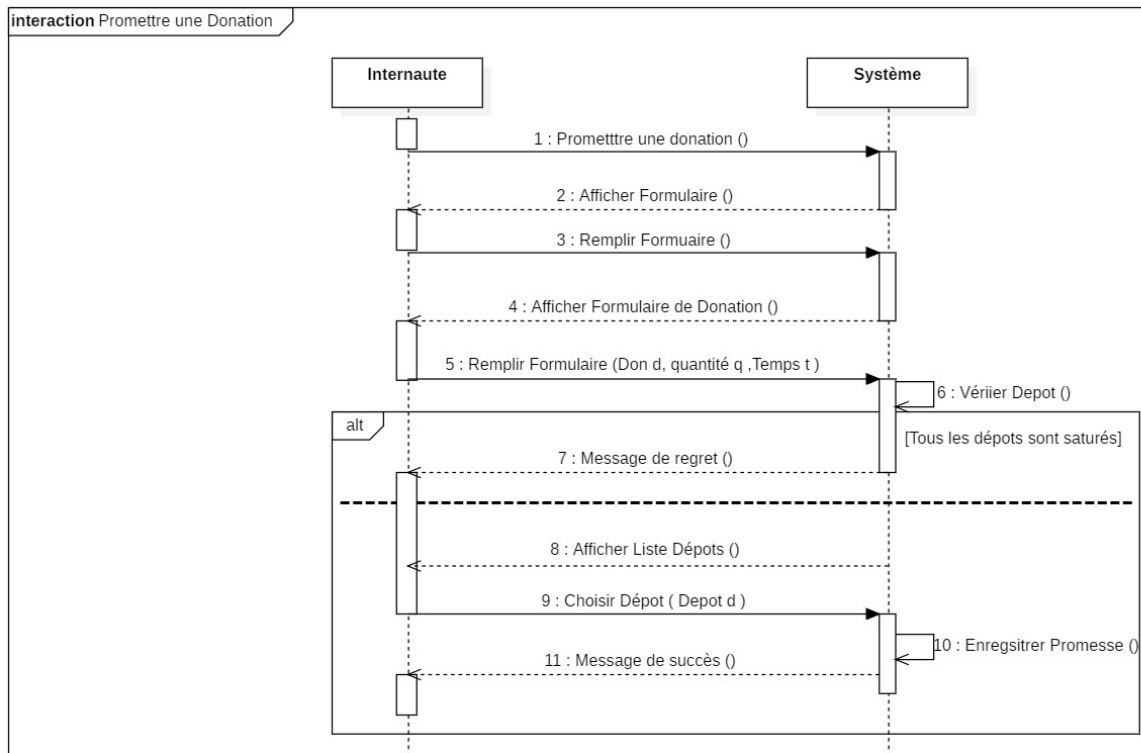


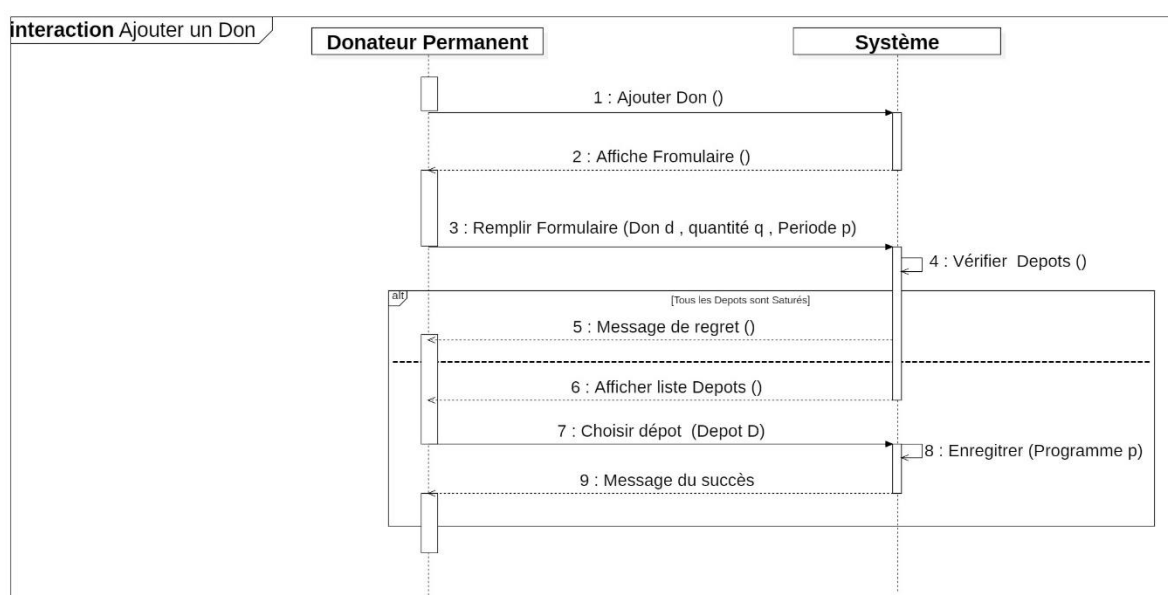
Figure 10 Diagramme de séquence système "Promettre une donation"



### 3.4 Le cas d'utilisation « Ajouter un don » :

Propriétés	Description
<b>Nom du cas</b>	Ajouter un Don.
<b>Acteurs</b>	Donateur permanent.
<b>Pré Conditions</b>	S'authentifier.
<b>Post Condition</b>	Enregistrement du programme dans le système.
<b>Scénario nominale</b>	<p>1/ L'utilisateur clique sur le bouton « Ajouter Don ».</p> <p>2/ Le Système demande à l'utilisateur d'entrer le type de don, sa quantité et la façon périodique</p> <p>3/ L'utilisateur entre ses choix.</p> <p>4/ Le système affiche à l'utilisateur la listes des dépôts disponibles pour sa demande.</p> <p>5/ L'utilisateur sélectionne le dépôt voulu.</p> <p>6/ Le système enregistre le programme et affiche à l'utilisateur un message de succès.</p>
<b>Scénario d'exception</b>	<p>4. e.1/Tous les dépôts sont saturés.</p> <p>4. e.2/Le système affiche un msg de regret.</p>

*Tableau 5 Fiche descriptive " Ajouter don"*



*Figure 11 Diagramme de Séquence Système "Ajouter un Don"*

### 3.5 Le Cas d'utilisation « Etablir une demande » :

Propriétés	Description
<b>Nom du cas</b>	Etablir une Demande.
<b>Acteurs</b>	Nécessiteux.
<b>Pré Conditions</b>	S'authentifier.
<b>Post Condition</b>	Enregistrement du la demande dans le système.
<b>Scénario nominale</b>	<p>1/ L'utilisateur clique sur le bouton « Faire une Demande ».</p> <p>2/ Le Système demande à l'utilisateur d'entrer le type de don souhaité en spécifiant la quantité.</p> <p>3/ L'utilisateur passe sa commande.</p> <p>4/ Le système affiche au nécessiteux la liste des dépôts disponible pour sa demande pour recevoir sa commande.</p> <p>5/ L'utilisateur choisit le dépôt.</p> <p>6/ Le système enregistre la demande et envoie un Email au nécessiteux en l'informant la date et l'heure de livraison.</p>
<b>Scénario alternatif</b>	<p>6.a.1/Le don choisi n'est pas disponible dans le dépôt choisi.</p> <p>6.a.2/Effectuer 6 du scénario nominal + notifier le gestionnaire de dépôt sur le transfert.</p> <p>6.b.1/Don demandé non disponible.</p> <p>6.b.2/ Le système informe le nécessiteux que le don demandé n'est pas disponible à présent et lui demande s'il veut que sa demande soit en attente.</p> <p>6.b.3/ Le nécessiteux clique sur « OUI ».</p> <p>6.b.4/ Le système enregistre la demande et envoie un Email au nécessiteux en l'informant qu'il sera notifié dès que son don est disponible.</p> <p>6.b.3.1/Le nécessiteux clique sur « NON ».</p> <p>6.b.3.2/Retourner vers 2.</p>

Tableau 6 Fiche descriptive "Etablir une demande"

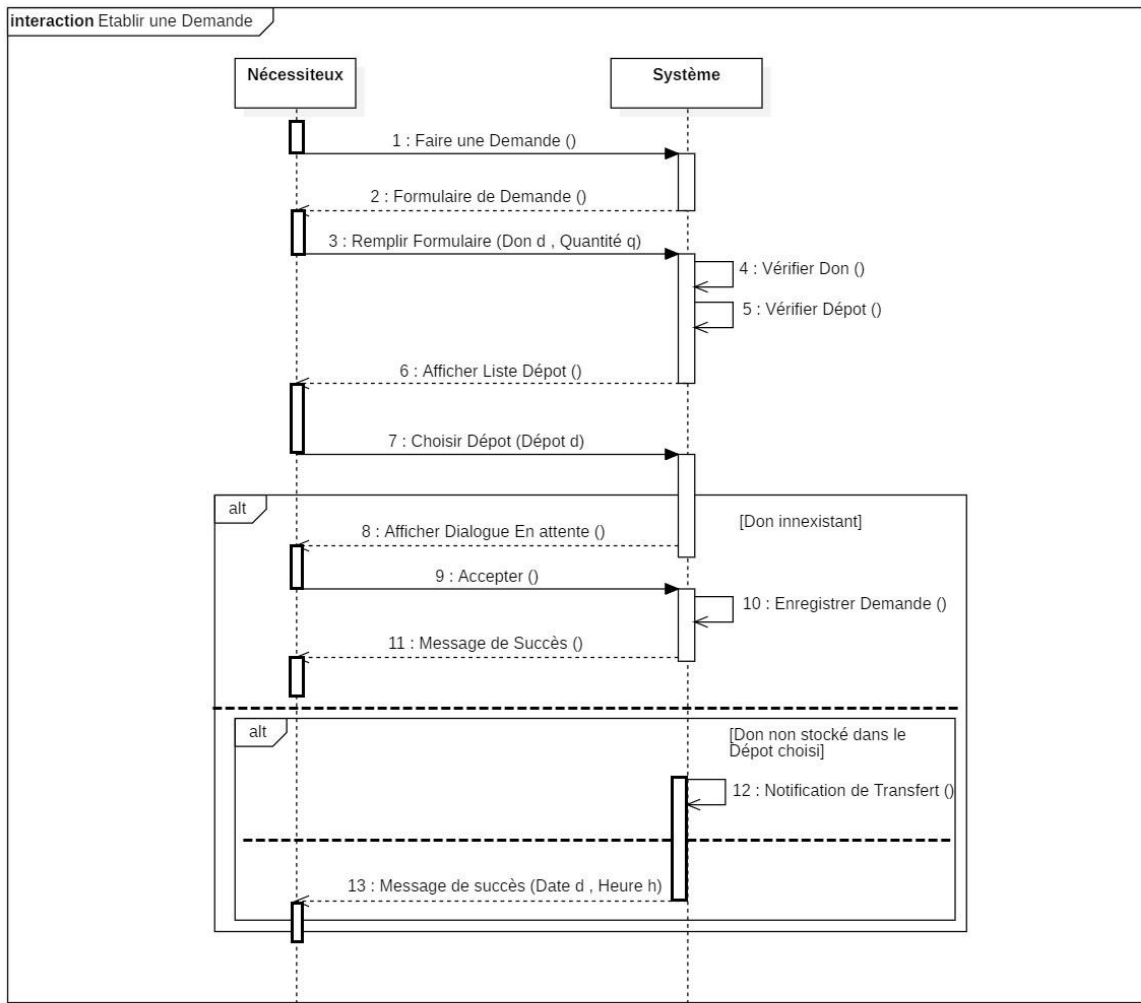


Figure 12 Diagramme de Séquence Système "Etablir une demande"

### 3.6 Le Cas d'utilisation « Valider Entrée » :

Propriétés	Description
<b>Nom du cas</b>	Valider entrée.
<b>Acteurs</b>	Gestionnaire de Dépôt.
<b>Pré Conditions</b>	-S'authentifier. -Don ajouté déjà par un donateur. -Présence du donateur et de son don.
<b>Post Condition</b>	Augmentation de la quantité d'un type de don dans le stock.
<b>Scénario nominale</b>	1/ L'utilisateur clique sur le bouton « Afficher les Dons ». 2/ Le Système affiche à l'utilisateur la liste des dons établis par les donateurs. 3/ L'utilisateur clique sur le bouton « Valider » d'un des dons. 4/ Le Système enregistre la commande, modifie la quantité du don choisi, affiche un message de succès.
<b>Scénario alternatif</b>	3.1/ L'utilisateur clique sur le bouton « Refuser » d'un des dons. 3.2/ Le Système enregistre le refus, notifie le donateur en l'envoyant un e-mail d'avertissement. 3.2.1/Le nombre d'avertissement atteint son max. 3.2.2/Le système notifie l'admin et envoie un e-mail au donateur en l'informant qu'il est banni.
<b>Scénario d'exception</b>	4.1/Le dépôt est saturé. 4.2/Le système affiche un message de regret et le retourne vers sa page personnelle.

*Tableau 7 Fiche Descriptive "Valider Entrée"*

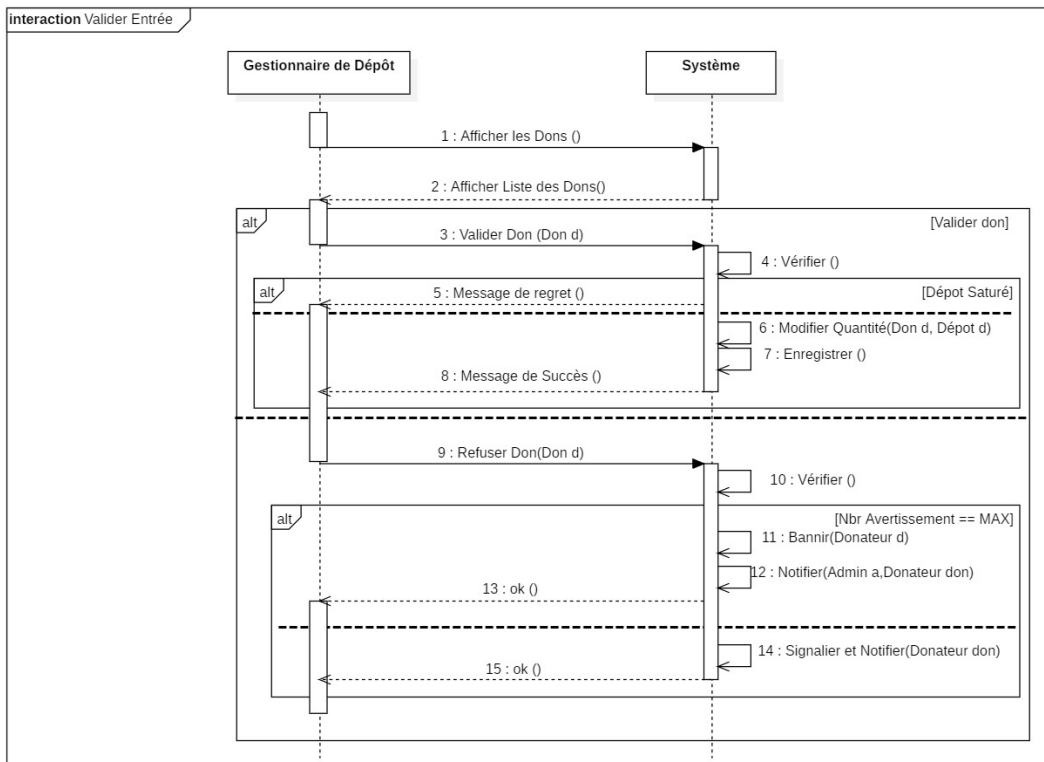


Figure 13 Diagramme de Séquence Système "Valider Entrée"

### 3.7 Le Cas d'utilisation « Valider Sortie » :

Propriétés	Description
Nom du cas	Valider sortie.
Acteurs	Gestionnaire du Dépôt.
Pré Conditions	-S'authentifier .-Demande déjà établit par un nécessiteux. -Présence du nécessiteux.
Post Condition	Diminution de la quantité d'un type de don dans le stock.
Scénario nominale	1/ L'utilisateur clique sur le bouton « Afficher Demandes ». 2/ Le Système affiche à l'utilisateur la liste des demandes des nécessiteux. 3/ L'utilisateur clique sur le bouton « Valider » d'une des demandes. 4/ Le Système enregistre la livraison, modifie la quantité du type de don choisi, affiche un message de succès.
Scénario alternatif	3.1 L'utilisateur clique sur le bouton « Refuser » d'une des demandes. 3.2/ Le Système enregistre le refus, notifie le nécessiteux en l'envoyant un e-mail d'avertissement. 3.2.1/Le nombre d'avertissement atteint son max. 3.2.2/Le système notifie l'admin et envoie un e-mail au nécessiteux en l'informant qu'il est banni.
Scénario d'exception	4.1/Quantité de don non disponible. 4.2/Le système affiche un message de regret et le retourne vers sa page personnelle.

*Tableau 8 Fiche Descriptive "Valider Sortie"*

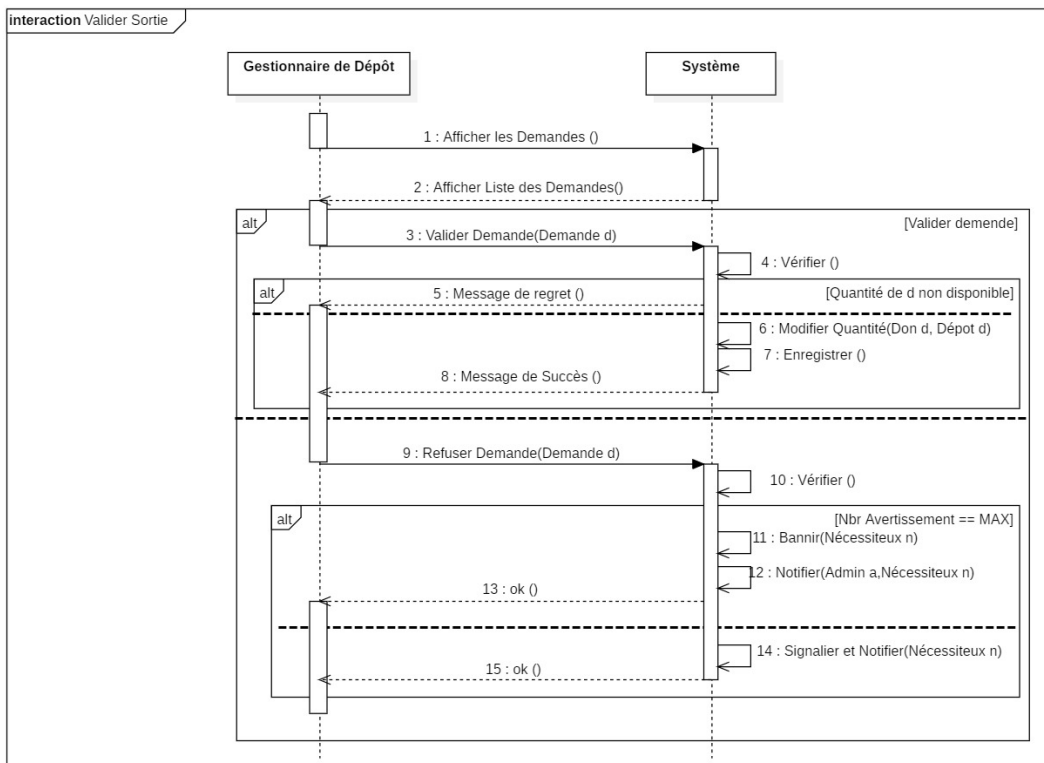


Figure 14 Diagramme de Séquence Système "Valider Sortie"

### 3.8 Le Cas d'utilisation « Effectuer un Transfert » :

Propriétés	Description
Nom du cas	Effectuer un Transfert.
Acteurs	Gestionnaire du Dépôt.
Pré Conditions	S'authentifier.
Post Condition	Transfert de la quantité d'un don du dépôt vers un autre.
Scénario nominale	<p>1/ Le gestionnaire de dépôt clique sur le bouton « Effectuer un Transfert ».</p> <p>2/Le système demande à l'utilisateur de choisir le dépôt donneur et le dépôt receveur, le type de don et sa quantité.</p> <p>3/ L'utilisateur entre ses choix.</p> <p>4/ Le Système enregistre le transfert, modifie la quantité du type de don dans les 2 dépôts, affiche un message de succès.</p>
Scénario alternatif	<p>4.a.1/Quantité non disponible dans le dépôt donneur.</p> <p>4.a.2/Aller vers 2.</p> <p>4.b.1/Dépôt receveur saturé.</p> <p>4.b.2/Aller vers 2.</p>

Tableau 9 Fiche Descriptive "Effectuer un Transfert"

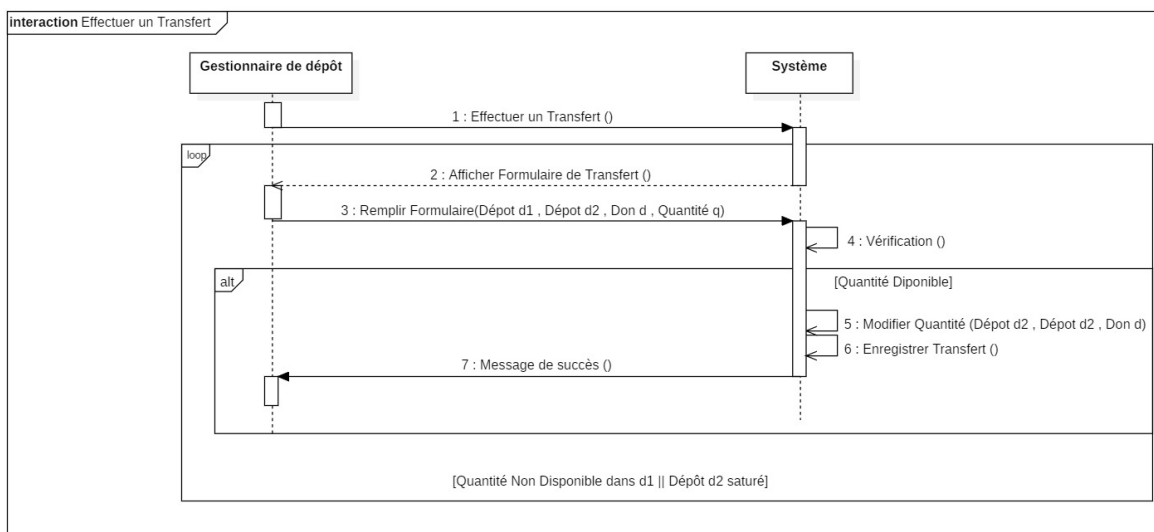


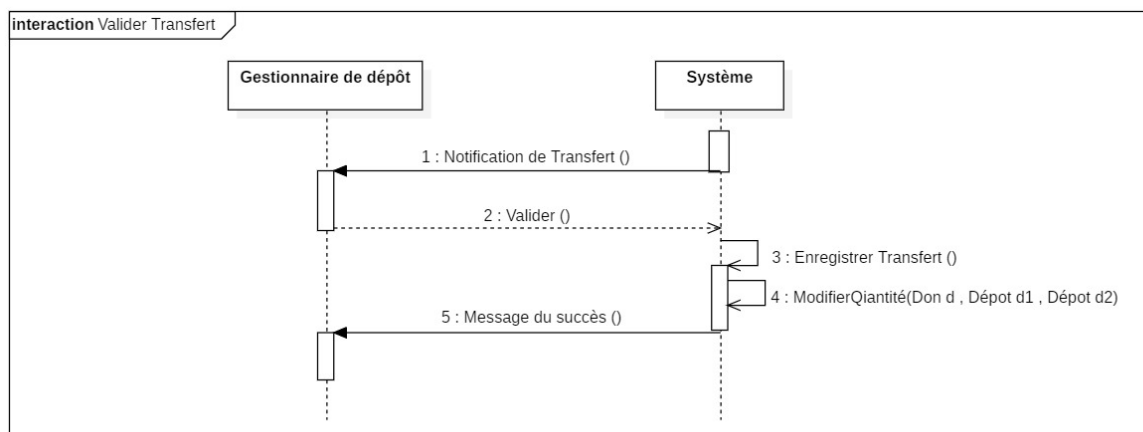
Figure 15 Diagramme de Séquence Système "Effectuer un Transfert"



### 3.9 Le Cas d'utilisation « Valider Transfert » :

Propriétés	Description
<b>Nom du cas</b>	Valider un Transfert.
<b>Acteurs</b>	Gestionnaire de Dépôt.
<b>Pré Conditions</b>	-S'authentifier. -Notification de Transfert.
<b>Post Condition</b>	Transfert de la quantité d'un don du dépôt vers un autre.
<b>Scénario nominale</b>	<p>1/ Le système notifie le gestionnaire de dépôt sur le transfert qui doit être effectuer.</p> <p>2/ L'utilisateur clique sur le bouton « Valider ».</p> <p>3/Le Système enregistre le transfert, modifie la quantité du type de don dans les 2 dépôts, affiche un message de succès.</p>

*Tableau 10 Fiche Descriptive "Valider Transfert"*

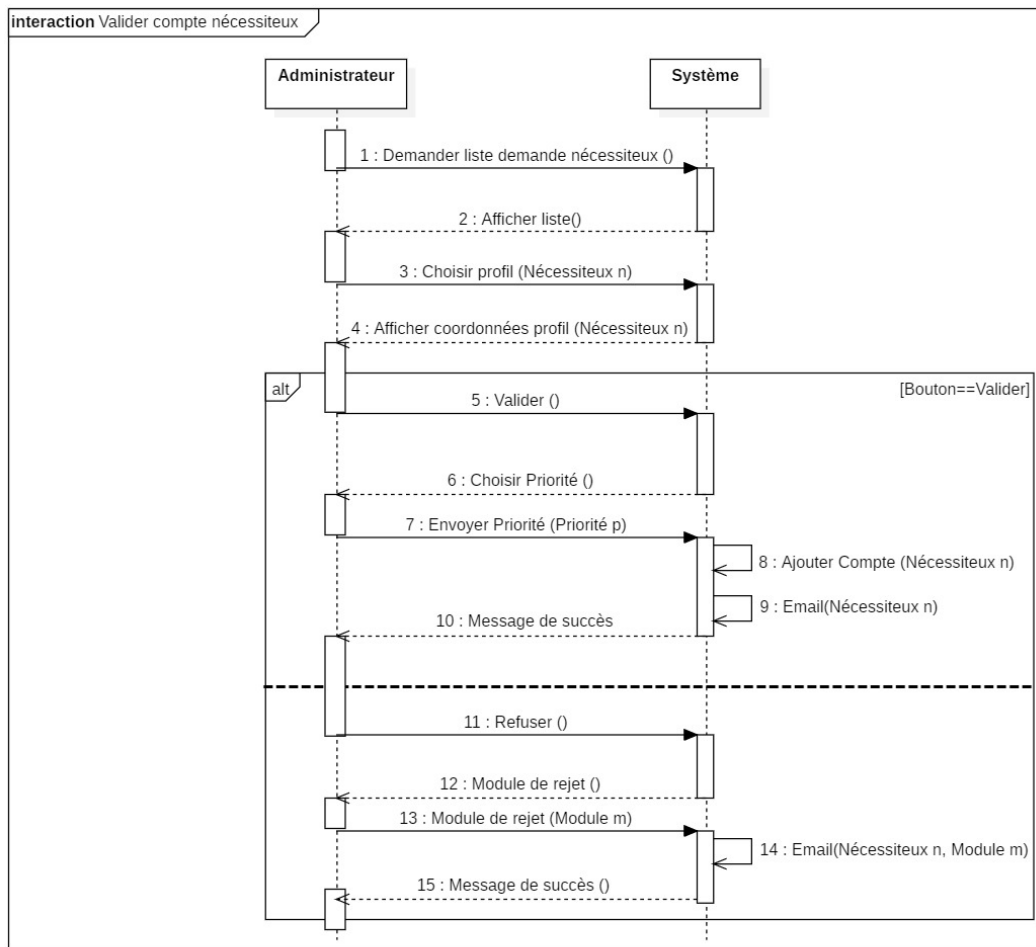


*Figure 16 Diagramme Séquence Système "Valider Transfert"*

### 3.10 Le Cas d'utilisation « Valider Compte nécessaire » :

Propriétés	Description
<b>Nom du cas</b>	Valider compte nécessaire.
<b>Acteurs</b>	Administrateur
<b>Pré Conditions</b>	-S'authentifier. -Demande établie par un nécessaire.
<b>Post Condition</b>	Compte nécessaire validé.
<b>Scénario nominale</b>	<p>1/ L'administrateur clique sur le bouton « Afficher liste de demande des nécessaires ».</p> <p>2/Le système affiche la liste des nécessaires en attente d'adhésion.</p> <p>3/ L'administrateur clique sur le statut de profil d'un des nécessaires.</p> <p>4/ Le Système affiche les coordonnées du profil sélectionné.</p> <p>5/L'administrateur clique sur le bouton « Valider ».</p> <p>6/Le système demande de sélectionner la priorité du nécessaire (A_B_C).</p> <p>7/L'administrateur choisi sa priorité.</p> <p>8/Le système valide le compte du nécessaire et lui envoie un email de confirmation.</p>
<b>Scénario alternatif</b>	<p>5.1/L'administrateur clique sur le bouton « Refuser ».</p> <p>5.2/Le système envoie un email d'échec au nécessaire.</p>

*Tableau 11 Fiche descriptive "Valider compte nécessaire"*



*Figure 17 Diagramme Séquence Système "Valider Compte nécessaireux"*

### 3.11 Le Cas d'utilisation « Annuler inscription » :

Propriétés	Description
Nom du cas	Annuler Inscription.
Acteurs	Nécessiteux & Donateur permanent.
Pré Conditions	S'authentifier.
Post Condition	Suppression du compte au sein du système.
Scénario nominale	<p>1/ L'utilisateur clique que le bouton « Annuler inscription ».</p> <p>2/Le système affiche un dialogue de confirmation.</p> <p>3/ L'utilisateur Valider l'annulation.</p> <p>4/ Le Système Supprime le compte et notifie l'admin.</p>

Tableau 12 Fiche Descriptive "Annuler inscription"

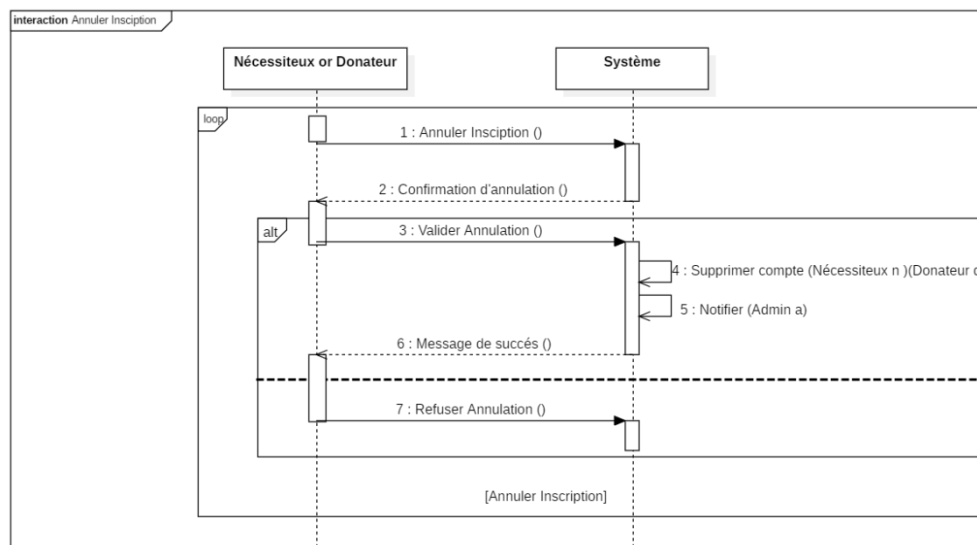


Figure 18 Diagramme Séquence Système "Annuler inscription"

## 4. Conclusion :

UML est un langage graphique utilisé dans la conception de logiciels. Ce langage permet d'échanger des informations entre des professionnels, chefs de projet et des utilisateurs. On a vu, à partir d'un exemple concret (association caritative) quelques diagrammes qui permettent de définir le besoin des futurs utilisateurs.

Ce qui nous pousse à avancer sur ce projet est que le processus UP nous permet fortement de revenir en arrière en cas de changement ou de redéfinition de quelques contraintes du système.

Dans ce chapitre, nous avons spécifié les besoins des utilisateurs de notre application en passant par les différentes étapes de la méthode Pascal Roque. Dans le chapitre suivant nous passerons à l'analyse et la conception en élaborant les différents diagrammes de classe.

# Chapitre 2 : Analyse et Conception

## 1. Introduction

Dans ce chapitre nous mettrons en évidence le coté conceptuel de notre application qui constitue une étape fondamentale qui précède l'implémentation, et qui permet de détailler les différents diagrammes et scénarios à implémenter dans la phase suivante. Ceci permettra une meilleure compréhension de l'application. Nous avons utilisé une démarche générale basé sur Pascal roques qui utilise le langage UML. Dans cette démarche, les diagrammes UML utilisés sont les suivant : Diagramme de cas d'utilisation, diagrammes de séquence et les diagrammes de classes sont modélisés à l'aide de l'outil StarUml.

## 2. Diagramme de classe :

Après l'achèvement du chapitre 1 (Etude préliminaire et spécification des besoins), on a pu extraire ce diagramme de classe représentant les différentes entités du système.

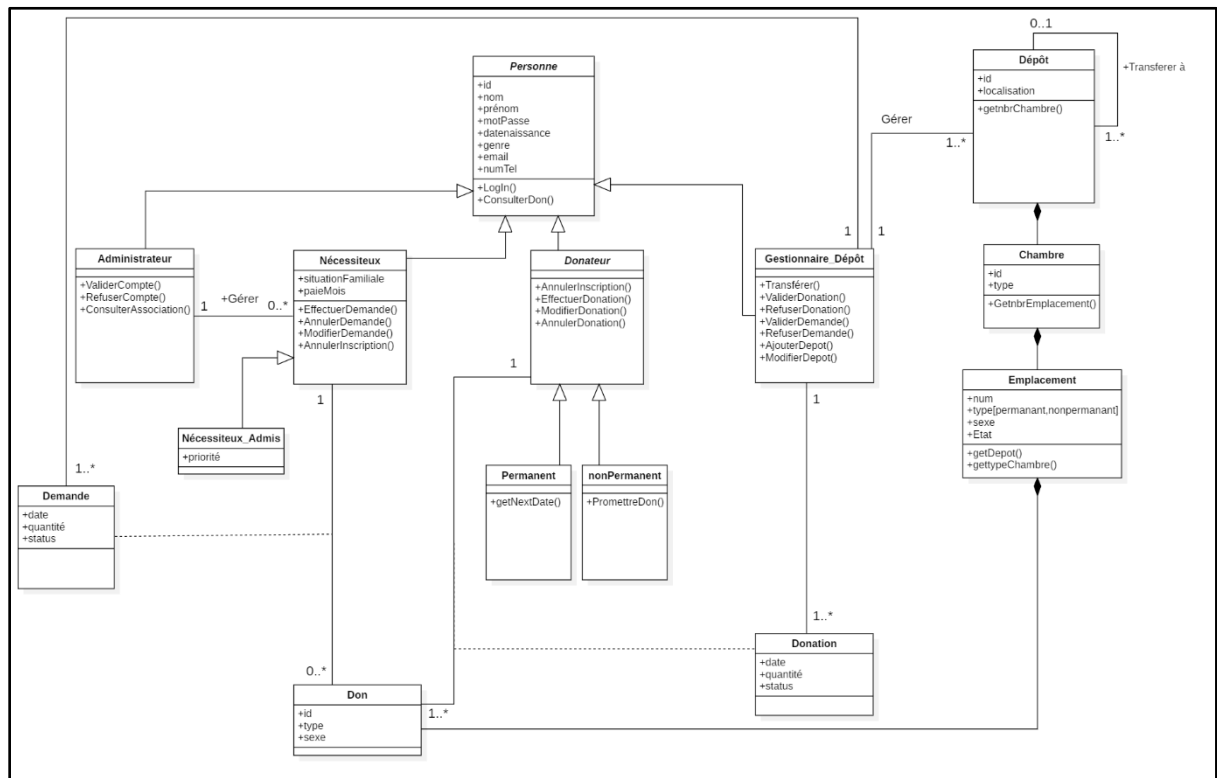


Figure 19 Modèle de Domaine

### 3. Modèle de diagramme d'activité :

Dans la phase de conception, les diagrammes d'activités sont particulièrement adaptés à la description des cas d'utilisation. Plus précisément, ils viennent illustrer et consolider la description textuelle des cas d'utilisation. Ils permettent de spécifier des traitements a priori séquentiels et offrent une vision très proche de celle des langages de programmation impératifs comme C++ ou Java.

On va donc représenter 2 scénario principale dans notre système qui regroupes entre les différents acteurs du système.

- **Ajouter une donation** par un donateur permanent.
- **Demander un don** par un nécessiteux.

#### 3.1 Donateur :

Ici le donateur n'est pas encore inscrit dans le système.

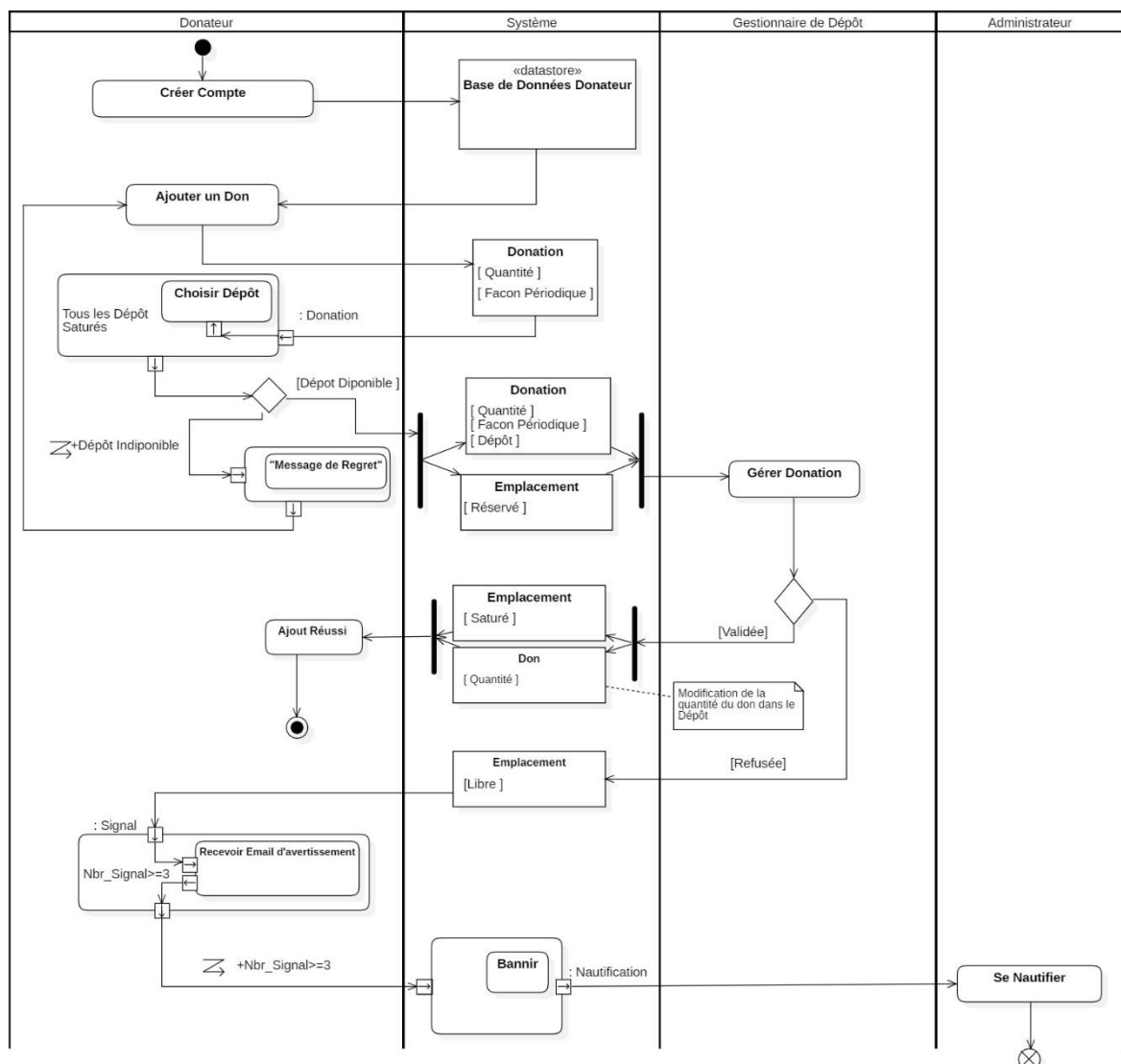


Figure 20 Diagramme d'activité Scénario "Donateur"

## 3.2 Nécessiteux :

Première interaction d'un nécessiteux dans le système.

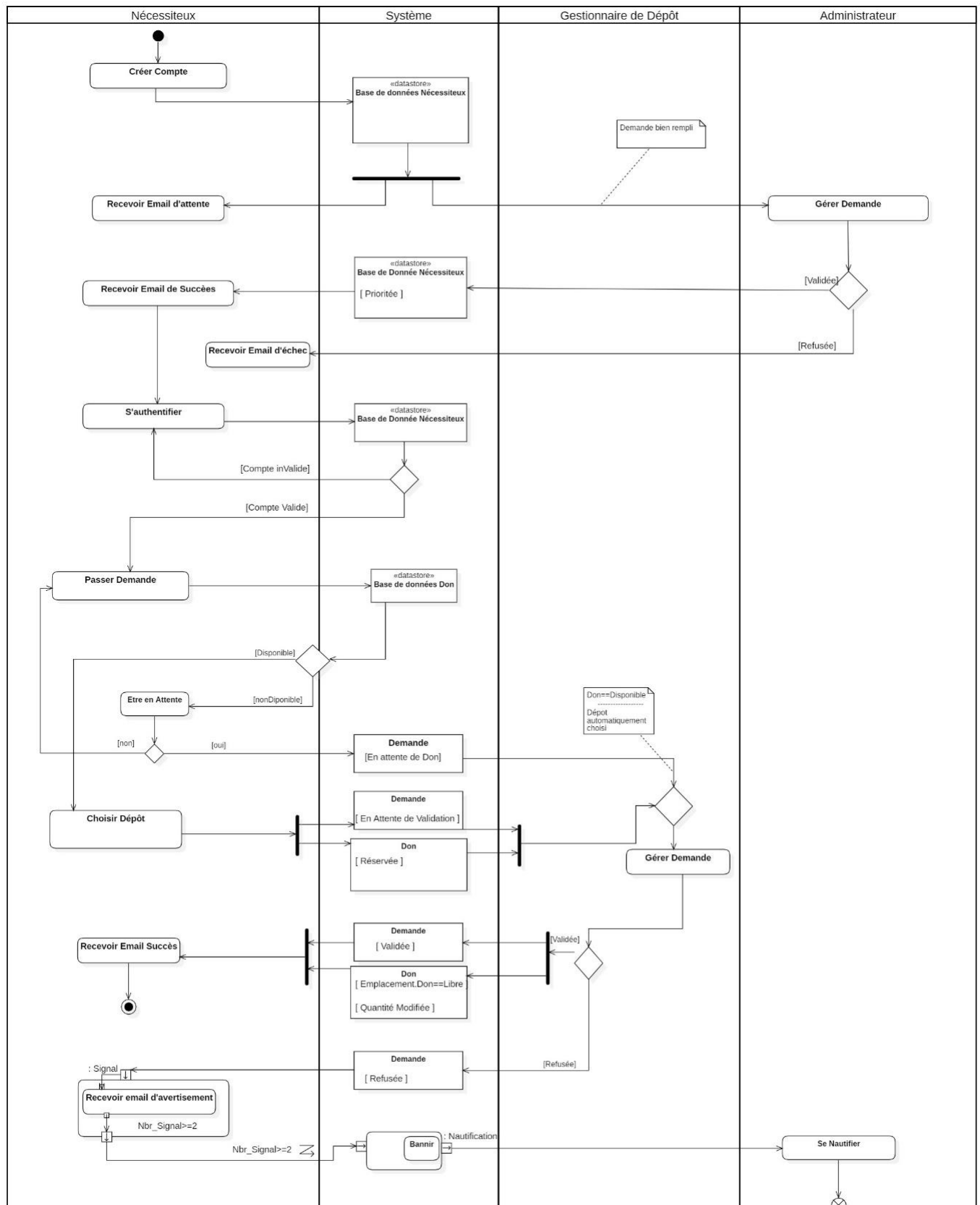


Figure 21 Diagramme d'activité Scénario "Nécessiteux"



## 4. Description de la base de données :

Le diagramme de classe qu'on a conçu est un schéma qui représente les classes et les différentes relations entre celles-ci (généralisation, association, composition...). Cependant un passage vers un modèle relationnel nécessite l'établissement des règles de passages conçu par des concepteurs avant. Mais ici, il y a une question qui se pose, pourquoi utiliser un modèle relationnel dans notre base de données ? pourquoi pas rester au modèle objet établis dans le diagramme de classe ?

Les principaux avantages des bases de données relationnelles sont qu'elles permettent aux utilisateurs de facilement **classer** et **stocker** des données qui peuvent ensuite être **interrogées** et **filtrées** pour extraire des informations spécifiques pour les requêtes. Les bases de données relationnelles sont également faciles à étendre et ne dépendent pas de l'organisation physique.

### 4.1 Règles de passage :

Voilà quelques règles employées dans notre cas :

#### 4.1.1 Classe Association :

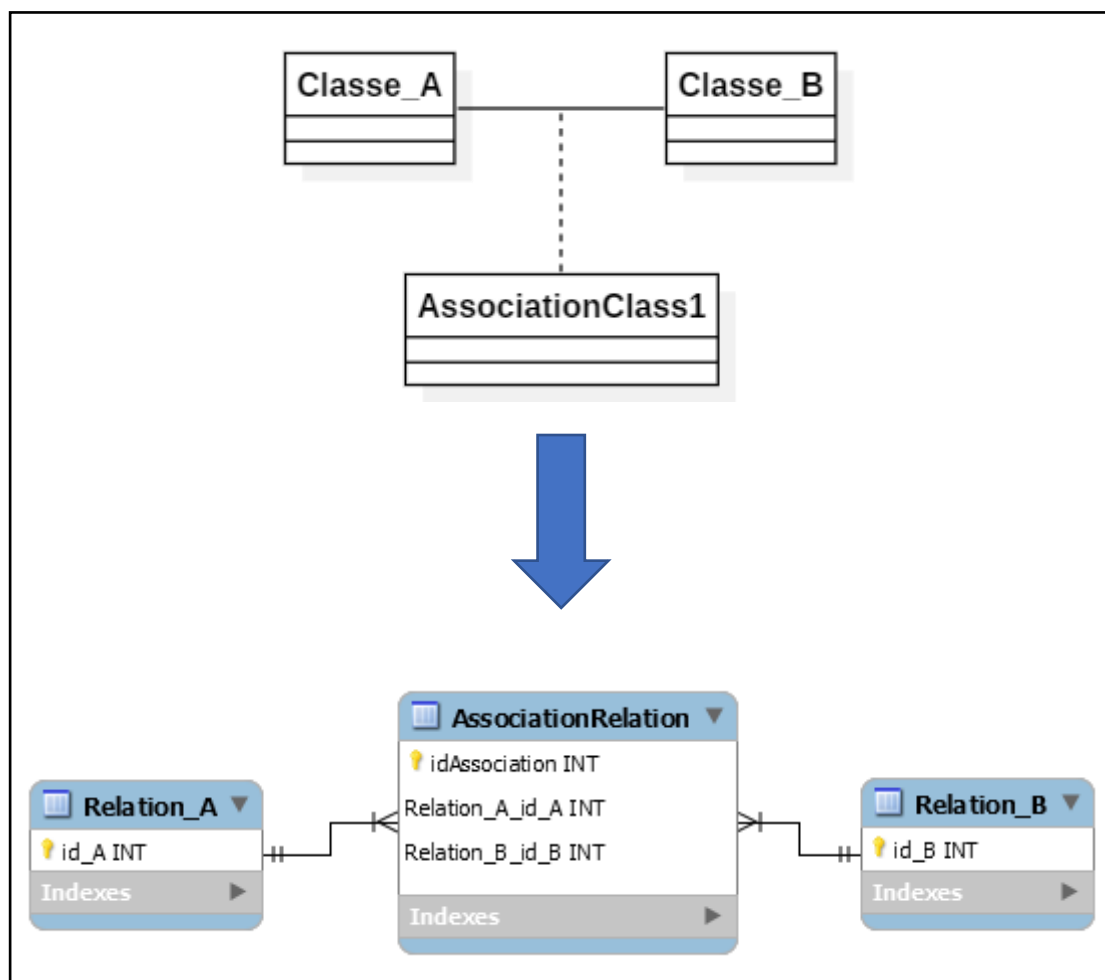


Figure 22 Passage "classe association"

### 4.1.2 Composition :

Une relation A se compose d'une relation B ou plus (cardinalité = 1...n).

Une relation B existait si seulement si un tuple de la relation A existait et n'appartient évidemment qu'à lui (cardinalité = 1).

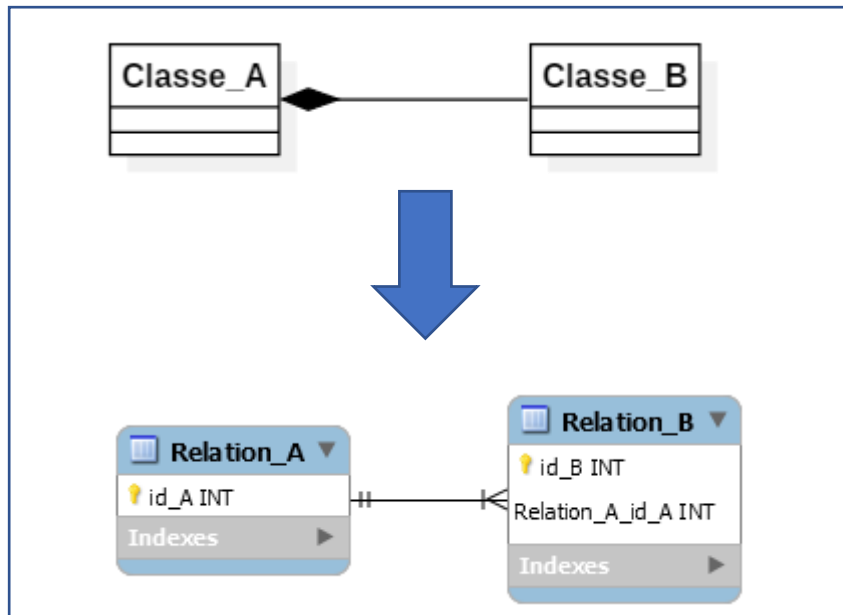


Figure 23 Passage "Composition"

### 4.1.3 Généralisation :

Une relation A peut être une relation B ou ne le pas être (cardinalité = 0...1).

Une relation B à son tour peut être qu'une seule fois si un tuple dans la relation A existait (cardinalité = 1).

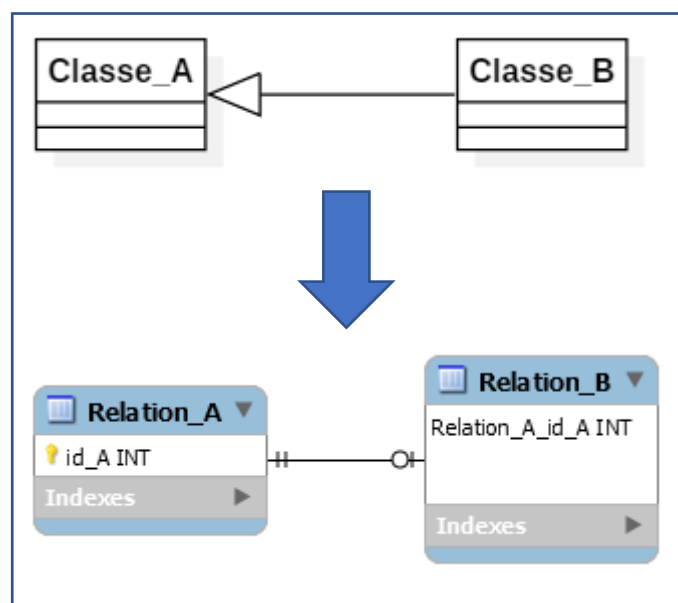


Figure 24 Passage "Généralisation"

## 4.2 Schéma de la base de données :

Suivant ce qui a précédemment été expliqué, l'élaboration d'un modèle relationnel nécessite le suivi des règles citées précédemment pour obtenir un schéma relationnel qui doit contenir des relations écrites en troisième forme normale.

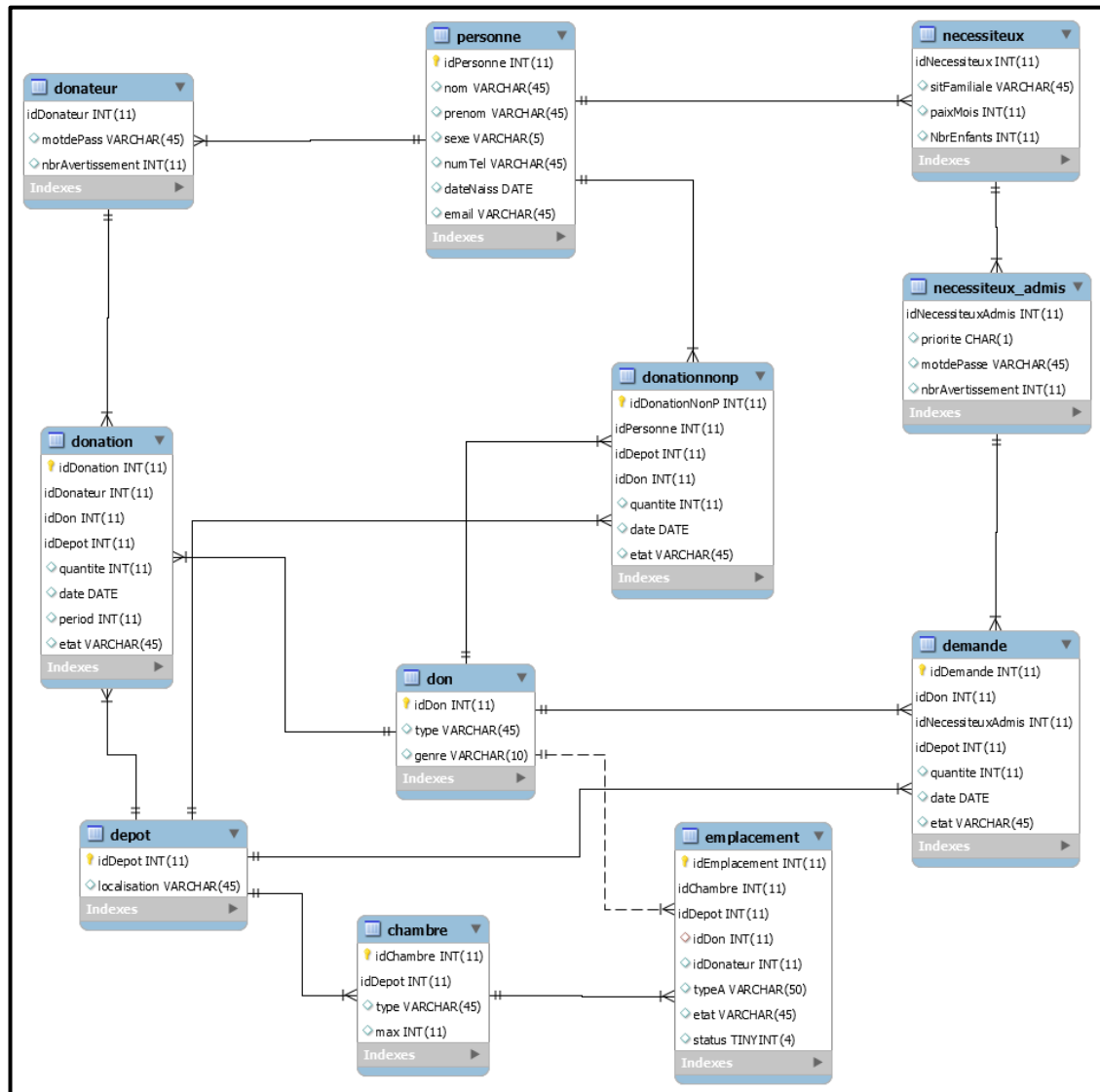


Figure 25 Schéma relationnel de la base de données

## 5. Conclusion :

L'achèvement de ce chapitre nous a permis d'avoir une vue non implémentée de notre système (classe, relation MySQL, traitement système ...).

Après le passage par les différentes étapes de développement, l'application a abouti à un logiciel fonctionnel qui répond globalement aux exigences de notre association. Dans le chapitre suivant on va mettre en œuvre notre conception, c'est-à-dire implémenter le site web et le rendre prêt à utiliser.

# Chapitre 3 : Implémentation

---

## 1. Introduction :

L'implémentation est le résultat de la conception pour implémenter le système sous formes de composants, c'est-à-dire, de code source, de scripts, de binaires d'exécutables et d'autres éléments du même type. Suite à la partie analyse et conception établit au chapitre précédent, nous allons présenter la partie implémentation et test du site et Application mobile, en commençant par présentation de l'environnement de travail et terminer par la description de quelques interfaces, pour au final présenter l'application dans son ensemble.

## 2. Environnement de Travail :

### • Plateforme matérielle :

Une plateforme matérielle représente les machines utilisées pour effectuer les tests et la réalisation du site et des applications mobiles avec les caractéristiques et la description de chaque machine.

Machine	Fabricant	Processeur	Modèle	RAM	Os
1	HP	Intel Core i7 2.20Ghz	Pavilion dv6	4Gb	Windows
2	Lenovo	Intel Core i5 2.40Ghz	Z50-70	6Gb	Windows

*Tableau 13 Ordinateur Utilisé*

### • Plateforme Logicielles :

Ci-dessous sont présentés tous les logiciels utilisés lors du développement :

- **Eclipse Javaee 64 bits** : Eclipse est un environnement de développement intégré (IDE) utilisé en programmation informatique. Dans ce projet, la version utilisée est 4.12.

- **MySQL Workbench** : MySQL Workbench est un outil visuel de conception de base de données intégrant le développement SQL. Dans ce projet, la version utilisée est 8.0.16.
- **Apache Tomcat** : Apache Tomcat (également appelé serveur Tomcat) implémente plusieurs spécifications Java EE, notamment Java Servlet, Java-Server Pages (JSP), Java EL et WebSocket, et fournit un environnement de serveur Web HTTP « pur Java » dans lequel le code Java peut être exécuté. Dans ce projet la version utilisée est 9.0.
- **Visual Studio Code** : Visual Studio Code est un éditeur de code source développé par Microsoft pour Windows, Linux et MacOS. Il est hautement personnalisable. Il permet aux utilisateurs de modifier le thème, les raccourcis clavier, les préférences et d'installer des extensions qui ajoutent des fonctionnalités supplémentaires. Dans ce projet la version utilisée est 1.37.1.
- **Font Awesome** : Font Awesome est un toolkit de polices et d'icônes basé sur CSS et LESS. Version utilisée est 5.10.2.
- **Adobe XD** : Adobe XD est un outil de conception d'expérience utilisateur à base de vecteur pour applications Web et applications mobiles, développé et publié par Adobe Inc. Version utilisée est 18.2.12.2.

### 3. Présentation des langages, technologies et Framework utilisées :

Lors de notre développement nous avons utilisé plusieurs langages de programmation et quelques nouvelles technologies dans le domaine de développement.

Ces langages sont :

- **HTML5** : HTML5 (HyperText Markup Language 5) est la prochaine révision majeure d'HTML (format de données conçu pour représenter les pages web). Cette version a été mise en développement en 2013. Nous utilisons HTML5 pour développer les différentes pages statiques du site web.
- **Bootstrap** : Bootstrap est un Framework CSS gratuit et à code source ouvert destiné au développement Web frontal réactif et premier mobile. Il contient des modèles de conception basés sur CSS et (éventuellement) JavaScript pour la typographie, les formulaires, les boutons, la navigation et d'autres composants d'interface.

- **JavaScript** : JavaScript un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client. C'est à dire au niveau du navigateur et non du serveur web.
- **jQuery** : jQuery est une bibliothèque JavaScript conçue pour simplifier la traversée et la manipulation d'arborescence DOM HTML, ainsi que la gestion d'événements, l'animation CSS et Ajax. C'est un logiciel gratuit et open-source utilisant la licence permissive MIT.
- **JSTL** : La bibliothèque de balises standard JSTL (JavaServer Pages) est un composant de la plate-forme de développement d'applications Web Java EE. Il étend la spécification JSP en ajoutant une bibliothèque de balises de balises JSP pour les tâches courantes, telles que le traitement de données XML, l'exécution conditionnelle, l'accès à la base de données, les boucles et l'internationalisation.
- **EL** : Le langage « Java Unifié Expression Language » est un langage de programmation à usage spécifique, principalement utilisé dans les applications Web Java pour incorporer des expressions dans des pages Web. Les rédacteurs de spécifications Java et les groupes d'experts des technologies de niveau Web Java ont travaillé sur un langage d'expression unifié qui a été initialement intégré à la spécification JSP 2.1 (JSR-245), puis spécifié par lui-même dans JSR-341, composant de Java EE. 7.
- **MySQL** : MySQL est un système de gestion de base de données relationnelle à source ouverte (RDBMS).
- **Gijgo** : Gijgo est un ensemble de contrôles javascript gratuits à usage commercial distribués sous licence MIT. Tous les widgets sont très performants et fonctionnent parfaitement avec Angular 2+, Bootstrap 4 et / ou jQuery. Ils sont conçus pour vous faire gagner du temps et évoluent avec votre processus de développement.
- **XML** : Le langage XML (Extensible Markup Language) est un langage de balisage qui définit un ensemble de règles de codage de documents dans un format lisible à la fois par ordinateur et par ordinateur. La spécification XML 1.0 du W3C et plusieurs autres spécifications connexes, toutes libres de normes ouvertes, définissent XML.

## 4. Règles de passage :

L'utilisation d'un SGBDR impose un changement de représentation entre la structure des classes et celle des données relationnelles. Les deux structures ayant une certaine analogie, le passage de l'objet au relationnel est possible suivant un ensemble de règles :

- **Choix du modèle MVC :**

- Les classes dialogues projetés vers des IHM (fichiers JSP et html).
- Les classes contrôles du modèle conceptuel projeté vers des classes java « Servlet ».
- Les classes métiers projetés vers les tables au niveau de la Base de données « Beans » grâce aux classes objets d'accès aux données (DAO).

- **Choix des servlets :**

- Puissance : Bénéficie des API noyau Java, (réseau, connectivité aux BD, invocation de méthode à distance, multithreading, ...) et de la plate-forme J2EE.
- Efficacité : Chargée une seule fois, c'est un objet qui reste en mémoire du serveur qui l'utilise par invocation de ses méthodes.
- Sûreté : Typage fort, exceptions, gestionnaire de sécurité Java.
- Orientées Objet : extensibilité des API, modularité.

- **Implémentation en SQL :**

Il est possible de traduire un diagramme de classe en modèle relationnel en suivant les règles de passages suivantes :

- Chaque classe UML devient une relation.
- Les attributs de la classe deviennent les attributs de la relation.
- Pour représenter une association 1 vers 1 entre deux relations, la clé primaire de l'une des relations doit figurer comme clé étrangère dans l'autre relation.
- Pour représenter une association 1 vers plusieurs, on procède comme pour une association 1 vers 1, excepté que c'est forcément la relation du côté plusieurs qui reçoit comme clé étrangère la clé primaire de la relation du côté 1.
- Pour représenter une association du type plusieurs vers plusieurs, il faut introduire une nouvelle relation dont les attributs sont les clés primaires des

relations en association et dont la clé primaire est la concaténation de ces deux attributs.

## 5. Présentation de l'application :

### • Modèle de l'application :

L'architecture client/serveur désigne un mode de communication entre plusieurs ordinateurs d'un réseau qui distingue un ou plusieurs postes clients du serveur. Le dialogue entre le client et le serveur se fait par échange de messages plutôt que par mémoire partagée, il est réalisé par échange de deux messages :

- Une requête (demande) du client pour l'exécution d'un service par le serveur.
- Une réponse envoyée par le serveur et qui contient le résultat du service. Cette application est basée sur ce type d'architecture, et elle comporte deux types d'application :
  - L'application web : Pour les clients, les employés, ainsi que les administrateurs de la société.
  - Les applications Android : Pour les clients et les agents.

### • Le Responsive Web design :

Le responsive Web design est une approche de conception Web qui vise à l'élaboration de sites offrant une expérience de lecture et de navigation optimales pour l'utilisateur quelle que soit sa gamme d'appareil (téléphones mobiles, tablettes, liseuses, moniteurs d'ordinateur de bureau).

Une expérience utilisateur "Responsive" réussie implique un minimum de redimensionnement (zoom), de recadrage, et de défilements multidirectionnels de pages.

Notre Site a été conçu avec cet approche, voici quelques exemples :

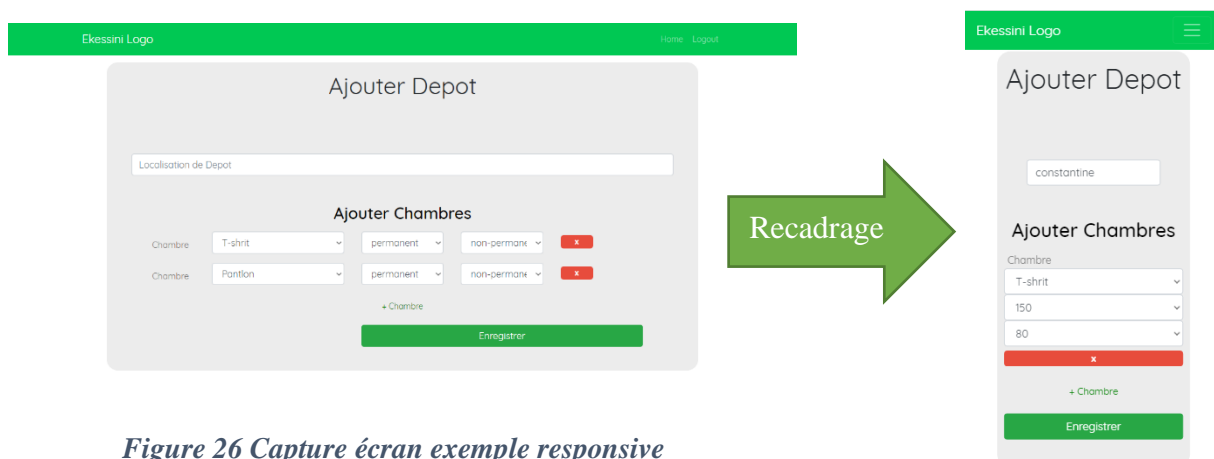


Figure 26 Capture écran exemple responsive



- **Capture écrans :**

## 5.1 Authentification :

Ekessini Logo Main Sign In

**SIGN IN**

Email tarekBoufar@gmail.com

Password \*\*\*\*\*

[Forgot Password?](#)  
[Dont have an Account ?](#)

Login

*Figure 27 Capture écran "Authentification"*

## 5.2 Profil Admin :

Ekessini Logo Home Log Out

Press F11 to exit full screen mode

[Dashboard](#) / Overview

**1 Necessiteux**  
voir les détails

**1 Necessiteux non-validé**  
Voir les détails

**comptes bloqués**  
voir les détails

[Top Donateurs](#) / Overview

#	Nom	Prenom	sexe	Telephone	Email	Donations
1	youcef	chabane	m	795875779	chabaneyoucef98@gmail.com	1

*Figure 28 Capture écran "Profil Admin"*

- Exemple de scénarios :

- Scénario « Créer Dépôt » :

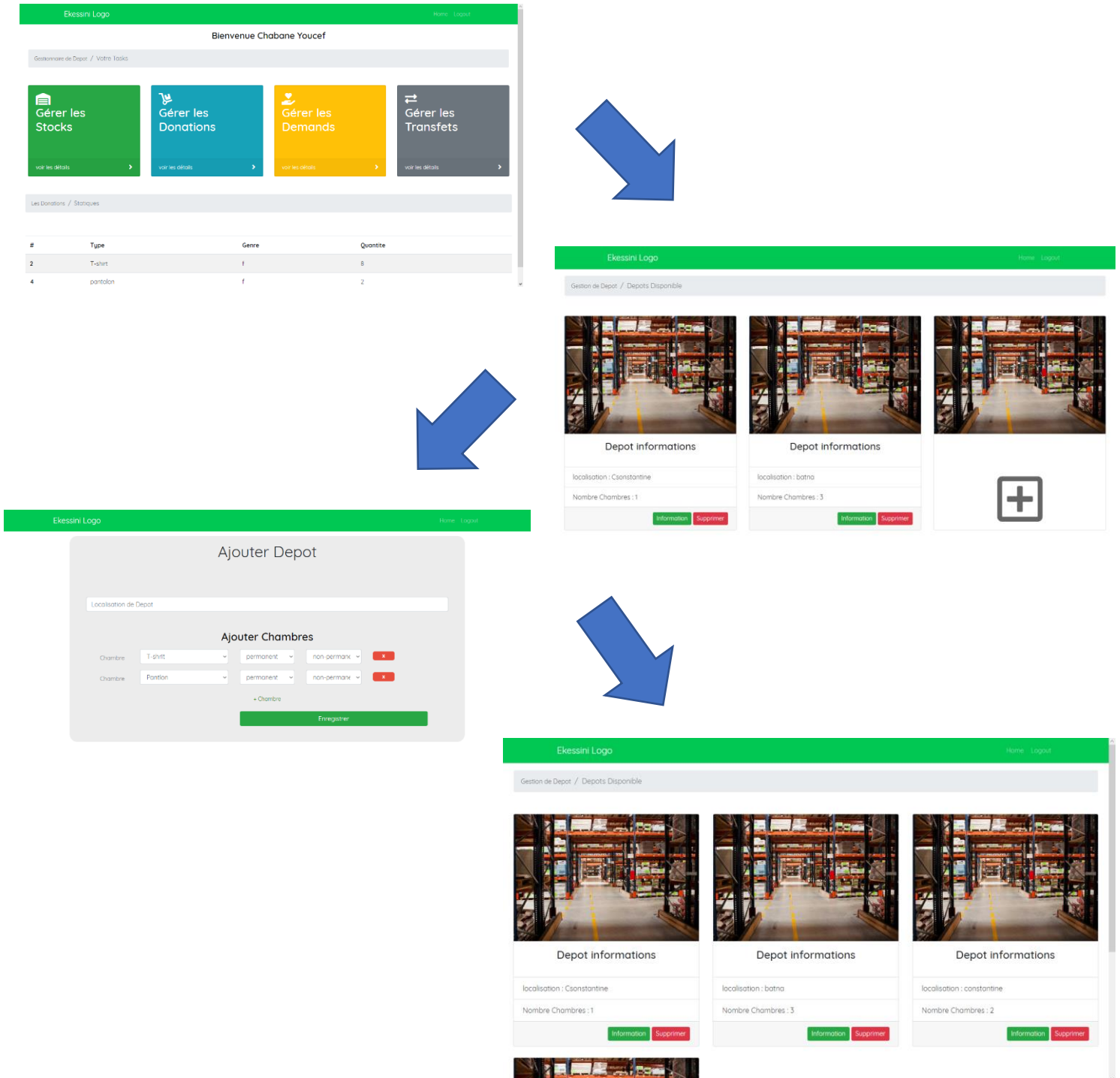


Figure 29 Scénario "Créer Dépôt"

- Scénario « Ajouter donation » :

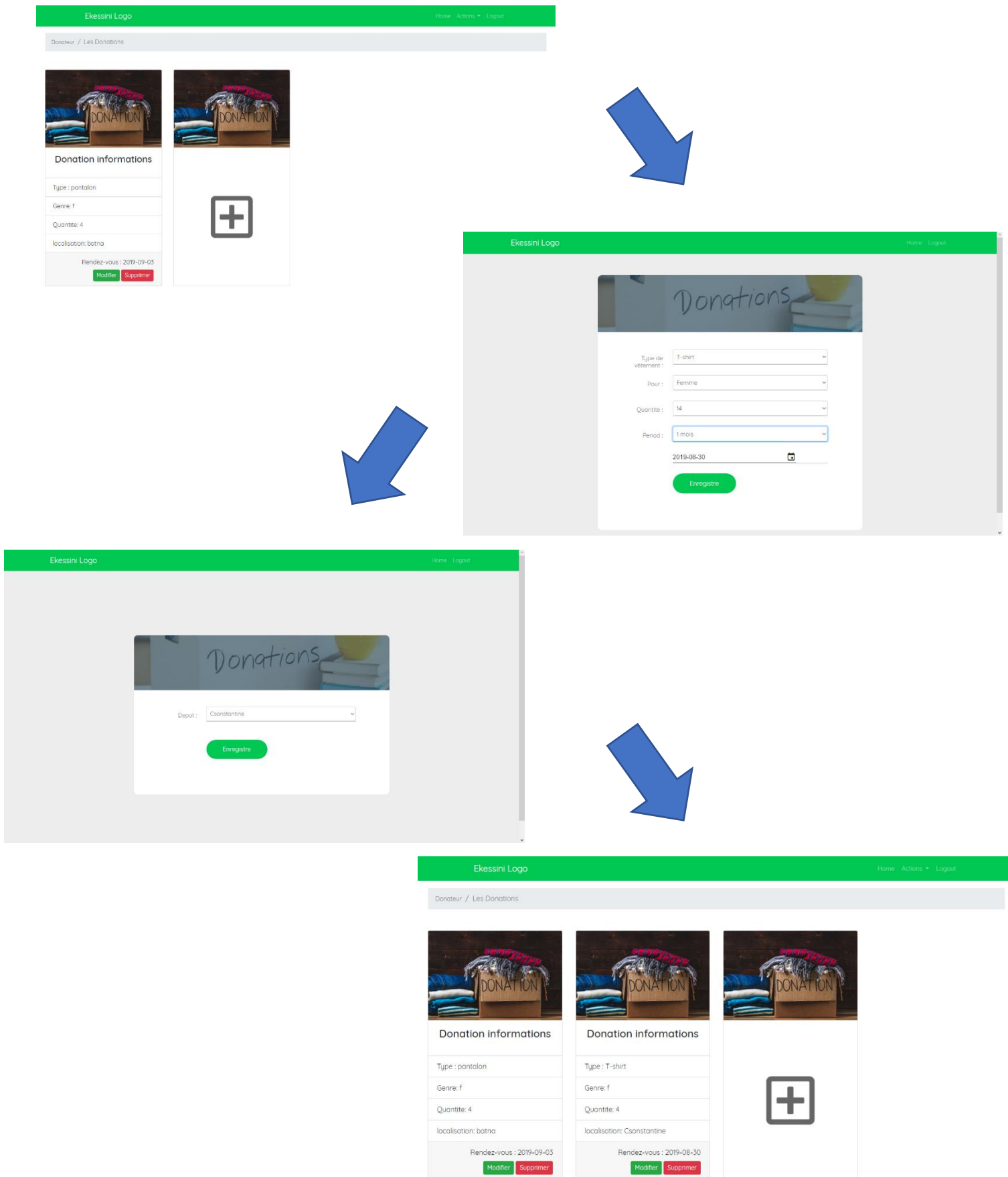


Figure 30 Scénario "Ajouter donation"

## 6. Conclusion :

Dans ce chapitre, nous avons présenté la structure du site, , les langages que nous avons utilisés, ainsi que les autres outils techniques et matériels, en terminant par quelques aperçus qui montrent les différentes parties de ce travail.

### Conclusion générale :

L'objectif de notre projet c'était de créer un site Web permettant de gérer les dons d'une association caritative, offrant différents services comme : l'ajouter des dons (entrées), demander une donation (sorties), créer des dépôts qui vont stocker les dons ...etc. Plusieurs technologies ont été nécessaires pour la réalisation de notre projet, on citera donc le langage HTML et JSP pour la réalisation des pages statiques, le langage JAVASCRIPT pour la partie dynamique, SQL pour l'élaboration des requêtes d'interrogation de la base de données, et enfin le logiciel Eclipse javaee comme environnement de développement. Après le passage par les différentes étapes de développement, l'application a abouti à un logiciel fonctionnel qui répond globalement aux critères imposés dans ce domaine. Le présent travail nous a permis d'acquérir des connaissances dans le domaine de la programmation web, et de renforcer nos connaissances en conception logicielle.

## Bibliographie :

- **Site :**

- Wikipédia l'encyclopédie libre on ligne ([www.wikipedia.org](http://www.wikipedia.org)).

Conception et analyse des besoins :

- OpenClassrooms ([www.openclassrooms.com/fr/](http://www.openclassrooms.com/fr/)).
- Developpez.com ([www.developpez.com/](http://www.developpez.com/)).

Implémentation :

- Bootstrap online documentation ([www.getbootstrap.com/](http://www.getbootstrap.com/)).
- Stack Overflow ([www.stackoverflow.com](http://www.stackoverflow.com)).
- Code Java ([www.codejava.net/java-ee](http://www.codejava.net/java-ee)).
- Coderanch ([www.coderanch.com](http://www.coderanch.com)).
- Tutorialpoint ([www.tutorialspoint.com](http://www.tutorialspoint.com)).

- **Livre :**
  - UML 2 de l'apprentissage à la pratique.
  - UML2 Par la pratique.
  - Apprenez à programmer en java.